Persian Magazine
Web Security Mgz .com

نفوذ در سرورهای وب

« بخش اول »

نفوذ در سرور Apache

تهیه کننده : امیر حسین شریفی info@websecuritymgz.com

تاریخ : 23 دی ماه سال 1382

منابع:

1- Hacking Exposed, Web Application

2- www.SRCO.ir

مقدمه

اولین چیزی که ممکن است از طریق آن برنامه های کاربردی مورد حمله واقع شوند ، سرورهای وبی می باشند که اینگونه برنامه ها روی آنها اجرا شده اند. هیچ برنامه کاربردی وب ، هرچند که خیلی مستحکم و امن هم باشد ، نمی تواند مدت زیادی روی یک سرور نا امن و آسیب پذیر دوام بیاورد.

در اینجا قصد بر این است که روی سرورهای وب معروفی همچون Apache ، IIS و Netscape بحث داشته باشیم. همچنین درباره آسیب پذیریهای ابتدایی و معمولی آنها و سپس معرفی و کار با چندین پوینده آسیب پذیری را برای شما بازگو کنیم. اگر عمری باقی ماند هم بر یک سری هم به حملات DOS [1] و تحلیل چگونگی انجام اینگونه حملات می زنیم.

خیلی صاف و ساده باید بگویم که اگر شما این بخش را مطالعه کنید در حال حاضر خیلی راحت می توانید به دنبال شکارهایی روی اینترنت باشید! اما همیشه یک کلاه سفید باشید ، یا به قول یکی از دوستان ، مانند یک سرخپوست از قبیله سوو وفادار به سنت ها باشید!!

آسیب پذیریهای متعارف

سالها قبل از اینکه امنیت سرورهای وب تجزیه و تحلیل شود ، ما فکر می کردیم که با انتخاب یک سرور خوب هیچگاه با مشکل آسیب پذیریهای بحرانی در چرخه حیات برنامه کاربردی خود مواجهه نخواهیم شد. اما در امروزه شما باید سرورهای وب خود را به صورت محتاطانه ای پیکر بندی کنید و همیشه آخرین Patch های آن را خریداری و نگهداری کنید و بعد از نصب آنها هنوز هم نمی توان خیلی مطمئن بود که همه چیز !OK است. در اینجا فقط تعداد بسیار محدودی از ضعفهای امنیتی سرورها بیان می شود و نحوه مقابله با آنها نیز گفته می شود ، ولیکن برای اطلاعات بیشتر دوستان باید به منابعی که در انتهای این دسته مقالات ذکر می شود مراجعه فرمایند.

1 - Denial Of Service

نفوذ در سرور Apache

آپاچی (Apache) یکی از متداولترین سرویس دهندگان وب بر روی اینترنت است . در مقایسه با سرویس دهنده وب مایکروسافت (IIS) ، آپاچی مسائل و مشکلات امنیتی کمتری را داشته ولی همچنان دارای آسیب پذیری خاص خود است .

علاوه بر وجود نقاط آسیب پذیر در ماژول ها و کد آپاچی (CA-2002-27 و CA-2002-17) ، تکنولوژی های CGI و PHP نیز دارای نقاط آسیب پذیری خاص خود بوده که ضعف های امنیتی آنان به سرویس دهنده وب نیز سرایت کرده است. در صورت وجود نقاط آسیب پذیر در سرویس دهنده آپاچی و یا عناصر مرتبط به آن ، زمینه تهدیدات زیر فراهم می گردد :

- **غیر فعال نمودن سرویس (DoS)**
- **نمایش و به مخاطره انداختن فایل ها و داده های حساس**
- **دستیابی به سرویس دهنده از راه دور**
- **به مخاطره افتادن سرویس دهنده (دستکاری و خرابی سایت)**

تمامی سیستم های یونیکس قادر به اجراء آپاچی می باشند . آپاچی بصورت پیش فرض بر روی تعداد زیادی از نسخه های یونیکس و لینوکس ، نصب می گردد .علاوه بر امکان فوق ، آپاچی را می توان بر روی میزبانی دیگر که از سیستم عاملی مختلف نظیر ویندوز استفاده می نماید نیز نصب نمود. این نوع از نسخه های آپاچی نیز می تواند دارای نقاط آسیب پذیر خاص خود باشد .

سایتهای تجارت الکترونیک فرض را بر این گذاشته اند که صفحات وب خود را به صورت دلخواه برای مشتریهای خود ایجاد کنند و قالب توجه خود را روی مشتری مداری قرار داده اند تا بتوانند مشتریها را به سایت خود جذب کنند به طوری که هر مشتری بر اساس علاقه مندی خود صفحات و موضوعاتی دلخواه را در سایت مشاهده کند و یا فلان صفحه را با فلان رنگ مشاهده کند. برای این منظور Apache نیاز به ماژولهایی دارد تا توانایی آن را برای طراحی و نمایش صفحات دینامیک بالا ببرد و همین ماژولها می باشند که Apache را در معرض خطر قرار می دهد. برای اینکه بهتر به این موضوع پی ببرید اجازه بدهید چند مثال و نمونه را با هم مرور کنیم.

لیست کردن دایرکتورها به وسیله Slash های طولانی

ترجمه URL های طولانی از میان ماژولهای mod_negotiate ، mod_dir و mod_autoindex باعث می شود که Apache محتوای دایرکتوریهای را نمایش دهد. این آسیب پذیری همان زمانی که Martin Kreamer نسخه Apache 1.3.19 را در مارس 2001 منتشر کرد ، نمایان بود. مفهومش خیلی ساده و پیش پا افتاده است ولی با کمی تقلا و تلاش می توان به وسیله آن بر سرور وب غلبه پیدا کرد. یک URL با تعداد زیادی از اسلش های دنباله دار ، به عنوان مثال :

/cgi-bin///

ممکن است لیست دایرکتوریهای درایوهای اصلی سیستم را نمایش دهد. تعداد واقعی اسلش ها متفاوت می باشد ولی می توان با نوشتن یک اسکریپت Perl خیلی ساده این حمله را پیاده سازی کرد. نکته ای که در اینجا باید گفته شود این است که بیشتر سرورهای Apache نمی توانند یک URL بزرگتر از 8000 کاراکتر را پردازش کنند.

اقدام متقابل در برابر اسلشهای طولانی

این ایراد در Apache 1.3.19 رفع شد . اگر چه این مشکل می توانست به وسیله پیکربندی بهتر و کاملتر Apache نیز رفع شود. ماژولهای mod_dir و mod_autoindex به صورت پیش فرض روی سرور Apache قرار دارند. این ماژولها که هر کدام می توانند لیست دایرکتوریها را نمایش بدهند به خاطر اینکه Apache کاربر پسند باشد ، روی سرور قرار گرفته اند ، و باید در زمان کامپایل کردن Apache آنها را حذف کرد. هیچ دلیلی وجود ندارد که محتوای دایرکتوریهای سرور وب خود را برای کاربر نهایی نمایش دهید. اسکریپتی که برای حل این مساله و حذف اینگونه ماژولها می توانید استفاده کنید در زیر آمده است.

[rohan apache]$ ./configure –disable-module=dir –disable-module=autoindex

توجه به این نکته ضروری می باشد که با حذف ماژول mod-dir در Apache در تمام درخواستهای وب کاراکتر اسلشهای پشت سر هم از قلم خواهند افتاد. هر چند که این موضوع تاثیر چندانی روی برنامه کاربردی ندارد.

دایرکتوری لیست شده Multiview

Apache بدون اجازه مدیر سرور ، با هر تلاشی که بخواهد به لیست دایرکتوری ها دست پیدا کند ، مخالفت می کند. متاسفانه ، یکی از جدید ترین قابلیت های Apache ، Multiviews ، که آسیب پذیری برای نشان دادن لیست دایرکتوریها می باشد توسط Kevin از سایت brasscannon.net در جولای 2001 به Bugtraq گزارش شد. این حمله می تواند به به وسیله مرورگر و یا از طریق دستور مستقیم برنامه netcat لیست دایرکتوری ها را نمایش دهد:

**[rohan]$ echo –e “GET /some_directory?M=D HTTP/1.0\n\n” | **
>nc 192.168.42.17 80

```
<!DOCUMENT HTML PUBLIC “-//W#C//DTD HTML 3.2 Final//EN”>
<HTML>
  <HEAD>
    <TITLE> Index of /some_directory </TITLE>
  </HEAD>
<BODY>
<H1>Index os /some_directory</H1>
<PRE><IMG SRC=”/icons/blank.gif” ALT=”    “><A HREF=”?N=A”>Name</A>
<A HREF=”?M=A”>Last modified</A>
<A HREF=”?S=A”>Size</A>
<A HREF=”?D=A”>Description</A>
<HR>
<A HREF=”/”>Parent Directory</A>    20-Oct-1998 08:45
<A HREF=”cgi-bin/”>cgi-bin</A>                20-Oct-1998 03:37
<A HREF=”messages/”>messages</A> 28-Oct-1998 01:36
<A HREF=”wwwboard.html”>wwwboard.html</A>     16-Apr-1998 19:44
<A HREF=”password.txt”>password.txt</A>    19-Apr-1998 12:01
<A HREF=”data.txt”>data.txt</A>               16-Apr-1998 02:53
<A HREF=”faq.html”>faq.html</A>               19-Apr-1998 04:09
</PRE><HR>
</BODY></HTML>
```

البته خروجی این دستور ، مقداری عوض شده است تا خوانا تر باشد. اما این یک مثالی از داده هایی می باشد که در یک دایرکتوری Apache می توان یافت. البته ما در آینده درباره فایلهای مهم این نوع سرور ها صحبت خواهیم کرد. برای حالا همان فایل Password.txt کافی می باشد. این آسیب پذیری بسیار مفید است زیرا لیست کاملی از دایرکتوریهای سایت را نشان می دهد.

راه مقابله با Multiview

اولین دفاع این است که اسناد درون دایرکتوری ریشه را پاک کنید. فایلهای غیر ضروری نباید در هر دایرکتوری ذخیره شود. یکه نکته برای توسعه دهندگان وب این است که هیچگاه داده

های قدیمی ، نسخه پشتیبان سایت ، و هر فایلی که مورد نیاز برنامه کاربردی نمی باشد ، نباید به وسیله مرور گر قابل دسترسی باشد. آسیب پذیری لیست کردن دایرکتوری ها وقتی ما را تهدید می کند که داده های حساس درون آنها وجود داشته باشد.

تزریق mod_auth_*sql

در آگوست 2001 ، RUS-CERT از دانشگاه Stuttgart روشی را منتشر کرد که چگونگی احراز هویت را بر پایه ماژولهای SQL توضیح می داد. تیک (') می توانست درون درخواستهای وب جاگذاری شود و همین به کاربران اجازه می داد که بتوانند دستورات SQL را شبیه سازی کنند که ساده ترین آن احراز هویت در یک سایت می باشد. البته این موضوع را ما در آینده به طور مفصل بحث خواهیم کرد.

راه مقابله با mod_auth_*sql

بسته mod_auth_*sql که استفاده می کنید را به روز رسانی کنید. البته لازم به ذکر است که پس از این کار باید Apache را دوباره راه اندازی کنید.

نحوه تشخیص آسیب پذیریهای سیستم

بمنظور آگاهی و کسب اطلاعات لازم در خصوص نحوه تشخیص آسیب پذیری سرویس دهنده وب آپاچی ، می توان از آدرس های زیر استفاده نمود :

- **در رابطه با Apache 1.3.x را می توان از آدرس**

http://www.apacheweek.com/features/security-13

- **برای Apache 2.0.x می توان از آدرس**

http://www.apacheweek.com/features/security-20

آدرس های اشاره شده ، دارای اطلاعات فنی لازم بمنظور نحوه تشخیص آسیب پذیری سیستم و پیشنهادات لازم در خصوص ارتقاء وضعیت امنیتی می باشند . استفاده از آدرس: http://httpd.apache.org/ نیز در این زمینه مفید است .

روشهای کلی حفاظت از سرویس دهنده وب Apache

بمنظور حفاظت یک سرویس دهنده وب آپاچی ، پیشنهادات زیر ارائه می گردد :

1- اطمینان از نصب آخرین patch ارائه شده

- در این رابطه می توان از آدرس http://httpd.apache.org/ بمنظور آگاهی از آخرین وضعیت نسخه ها و levels Patch استفاده نمود.

- بمنظور دستیابی به code Source اکثر نسخه های آپاچی، می توان از آدرس http://httpd.apache.org/download.cgi استفاده نمود.

- بمنظور آگاهی و دریافت آخرین Patch های ارائه شده می توان از آدرس http://www.apache.org/dist/httpd/patches/ استفاده نمود.

2- اطمینان از patching عناصر کلیدی سیستم عامل که آپاچی بعنوان مرجع از آنان استفاده می نماید .در این رابطه لازم است که صرفا" ماژول های ضروری بمنظور صحت عملکرد سرویس دهنده ، در آپاچی کمپایل گردند .لازم است به این نکته اشاره گردد که کرم mod_ssl (CA-2002-27) نمونه ای کامل در این زمینه بوده که از نقاط آسیب پذیر در OpenSSL (CA-2002-23) استفاده نموده است .

3- از اجرای آپاچی بعنوان ریشه ، اجتناب کنید و می بایست بدین منظور ، کاربر و یا گروهی خاص با حداقل مجوز ایجاد گردد. سایر پردازه های سیستم ضرورتی به اجراء تحت کاربر و یا گروه فوق را نخواهند داشت .

4- Chroot ، پتانسیلی است که باعث تعریف مجدد محدوده یک برنامه می گردد . در حقیقت chroot ، باعث تعریف مجدد دایرکتوری "ROOT" و یا "/" برای یک برنامه و یا یک Login session می گردد .chroot می تواند بعنوان یک لایه تدافعی استفاده گردد . مثلا" در صورتیکه فردی به کامپیوتر شما دستیابی پیدا نماید ، قادر به مشاهده تمامی فایل های موجود بر روی سیستم نخواهد بود . علاوه بر محدودیت فوق ، محدودیت هائی در خصوص اجرای برخی از دستورات نیز بوجود می آید. در این رابطه یک دایرکتوری با نام /chroot ، ایجاد می شود و تمامی سرویس های مورد نظر با یک انضباط خاص در آن مستقر می گردند . مثلا" سرویس دهنده آپاچی در chroot/httpd / قرار می گیرد. با توجه به موارد فوق ، می بایست آپاچی را در

یک محیط chroot اجراء نمود . درصورتیکه آپاچی بصورت chrooted اجراء و فعالیت خود را آغاز نماید ، امکان دستیابی آن به سایر بخش های موجود در ساختار دایرکتوری سیستم عامل و خارج از chroot وجود نخواهد داشت . بدین ترتیب یک لایه تدافعی مناسب در خصوص سوء استفاده های احتمالی ایجاد می گردد. بعنوان نمونه ، ممکن است یک shell فراخوانده شده و با توجه به اینکه bin/sky / در chroot قرار ندارد ، می تواند زمینه سوء استفاده احتمالی را فراهم نماید. لازم است به این نکته مهم نیز اشاره گردد که Chrooting آپاچی می تواند اثرات جانبی نامطلوبی را در ارتباط با CGI,PHP ، بانک های اطلاعاتی و سایر ماژول ها و یا ارتباطاتی که محیط سرویس دهنده وب بمنظور سرویس دهی به آنان نیازمند دستیابی به توابع کتابخانه ای خارجی است را بدنبال داشته باشد .روش های متعددی بمنظور chrooting وجود داشته و می بایست از مستندات نرم افزار مورد نظر ، بعنوان یک منبع اطلاعاتی مناسب در خصوص ارائه راهکارهای مربوطه ، استفاده گردد

5- بمنظور مدیریت یک سرویس دهنده وب ، لازم است فیدبک های لازم در خصوص فعالیت و کارآئی سرویس دهنده و سایر مسائلی که ممکن است یک سرویس دهنده با آنان برخورد نماید را اخذ کرد و در ادامه با آنالیز آنان تمهیدات لازم در خصوص مسائل موجود را بکار گرفت . سرویس دهنده آپاچی ، قابلیت ها و پتانسیل های انعطاف پذیری را در خصوص logging ارائه می نماید . بنابراین لازم است عملیات logging با دقت نظر بالا بصورت موثر و موشکافانه انجام گیرد تا امکان ردیابی هر نوع فعالیت امنیتی غیر مجاز و یا رفتار غیر منطقی سرویس دهنده ، فراهم گردد .پیشنهاد می گردد که با یک نظم خاص از اطلاعات موجود در فایل های لاگ ، آرشیو تهیه شود . بدین ترتیب ، امکان مدیریت فایل های لاگ و بررسی آنان فراهم خواهد شد. بمنظور آشنائی با فرمت های متفاوت لاگ می تواند از منابع زیر استفاده نمود :

- برای Apache 1.3.x از آدرس http://httpd.apache.org/docs/logs.html استفاده شود .

- برای Apache 2.0.x از آدرس http://httpd.apache.org/docs-2.0/logs.html استفاده شود .

در موارد متفاوتی و با توجه به شرایط پیش آمده ممکن است محتوی فایل های لاگ بتنهائی کافی نباشد . وضعیت فوق در مواردیکه از PHP ، CGI و یا سایر تکنولوژی های مبتنی بر اسکریپت استفاده می گردد ، تشدید می گردد و می توان بمنظور افزایش توان آنالیز یک تهاجم و سوءاستفاده از یک ضعف امنیتی ، اقدام به ثبت لاگ های مربوط به GET و POST نمود. لاگ نمودن عملیات مرتبط به GET و POST می تواند از طریق mod_Security صورت پذیرد.

ModSecurity یک سیستم تشخیص مزاحمین (detection Intruder) می باشد و پیشگیری های لازم در خصوص یک برنامه وب را ارائه می نماید . سیستم فوق به همراه سرویس دهنده وب مستقر می شود و یک پوشش امنیتی مناسب را در جهت پیشگیری از یک تهاجم در ارتباط با برنامه های وب فراهم می نماید . ModSecurity ، از سرویس دهنده آپاچی حمایت می نماید .

- http://www.modsecurity.org/
- http://www.securityfocus.com/infocus/17064.152.44.126%20152.44.126

6- PHP ، SSI،CGI و سایر اسکریپت ها

در این رابطه موارد زیر پیشنهاد می گردد :

- PHP,CGI,SSI و سایر زبان های اسکریپت را غیر فعال نمائید (مگر اینکه ضرورتی جدی در رابطه با آنان وجود داشته باشد).
- SSI یا Server Side Includes را که می تواند زمینه مساعدی به منظور سوء استفاده از سرویس دهنده فراهم کند و باعث اجرای کدهای ناخواسته گردد را غیر فعال نمائید .
- در صورتیکه ضروری است که از PHP,CGI,SSI و یا سایر زبان های اسکریپت استفاده گردد ، می بایست از SuEXEC استفاده شود. suEXEC ، امکان اجرای اسکریپت ها تحت آپاچی بهمراه یک User Id در مقابل یک Apache User Id را فراهم می نماید در حقیقت suEXEC این امکان را برای کاربران آپاچی فراهم می نماید که قادر به اجرای برنامه های SSI و CGI تحت یک User Id متفاوت نسبت به User Id مربوط به فراخوانی سرویس دهنده وب باشند.بدین ترتیب تهدیدات امنیتی کاهش و امکان نوشتن و اجرای برنامه های SSI و CGI اختصاصی نوشته شده توسط مهاجمان ، حذف خواهد شد . استفاده از suEXEC ،می بایست توام با آگاهی و دانش لازم باشد چراکه در صورت استفاده نادرست و یا عدم پیکربندی مناسب و شناخت نسبت به مدیریت setupid Root ، خود باعث بروز حفره های امنیتی دیگر خواهد شد.. در این رابطه و بمنظور آشنائی با نحوه عملکرد و استفاده از suEXEC می توان از آدرس های زیر استفاده نمود:

- برای Apache 1.3.x از آدرس زیر استفاده می شود :

http://httpd.apache.org/docs/suexec.html

-- برای Apache 2.0.x از آدرس زیر استفاده می شود:

http://httpd.apache.org/docs-2.0/suexec.html

- بررسی لازم در خصوص محتوی دایرکتوری cgi-bin و سایر دایرکتوری های شامل اسکریپت ها انجام و لازم است تمامی اسکریپت های پیش فرض نمونه ، حذف گردند.

ایمن سازی PHP

پرداختن به موضوع فوق با توجه به گستردگی مطالب از حوصله این مقاله خارج بوده و صرفا" به دو نمونه مهم در اینخصوص اشاره می گردد :

- غیر فعال نمودن پارامترهائی که باعث ارائه اطلاعات در HTTP header می گردد .
- حصول اطمینان از اجرای PHP در حالت safe

برای دریافت اطلاعات تکمیلی دراین خصوص می توان از آدرس http://www.securityfocus.com/printable/infocus/1706 استفاده نمود .

- استفاده از ماژولهای اضافه بمنظوربهبود وضعیت امنیتی. مثلا"ماژول mod_Security می تواند باعث حفاظت در مقابل Cross Site Scripting: XSS ، شود . برای آشنائی و مشاهده اطلاعات تکمیلی در این خصوص می توان از آدرس http://www.modsecurity.org/ استفاده نمود.
- ممیزی و بررسی اسکریپت ها برای نقاط آسیب پذیر شامل SQL Injection & XSS نیز حائز اهمیت است . در این رابطه می توان از ابزارهای متعددی استفاده نمود. نرم افزار Nikto (قابل دسترس در آدرس http://www.cirt.net/code/nikto.shtml) یکی از مناسبترین ابزارهای پویش و بررسی CGI است .