مرکز تخصصی آپا
دانشگاه صنعتی اصفهان
(آگاهی‌رسانی، پشتیبانی و امداد در حوزه شبکه)

تغییر مشخصات (هویت) کارگزار وب آپاچی

درجهت مقابله با حملات امنیتی

مرداد ۱۳۹۲

www.nsec.ir

۱. مقدمه

در این قسمت به روش هایی می پردازیم که باعث می شود که هویت وب سرور از نفوذگر ها پنهان بماند یا با دادن اطلاعات غلط آنها را گمراه کند.

۲. عوض کردن فیلد سرور در http header

۲.۱. عوض کردن نام ها در کد منبع

شما می توانید با عوض کردن دو قسمت در کد منبع آپاچی، هویت آن را عوض کنید. اولی در فایلی با نام httpd.h برای آپاچی ۱ و ap_release.h برای آپاچی ۲ است، جایی که نسخه ماکرو ها تعریف شده است:

```
#define SERVER_BASEVENDOR   "Apache Group"
#define SERVER_BASEPRODUCT  "Apache"
#define SERVER_BASEREVISION "1.3.29"
#define SERVER_BASEVERSION  SERVER_BASEPRODUCT "/" SERVER_BASEREVISION
#define SERVER_PRODUCT  SERVER_BASEPRODUCT
#define SERVER_REVISION SERVER_BASEREVISION
#define SERVER_VERSION  SERVER_PRODUCT "/" SERVER_REVISION
```

پیشنهاد می شود که شما فقط SERVER_BASEPRODUCT را تغییر دهید و version number را دستکاری نکنید. چرا که این فیلد بوسیله ماژول های دیگر داخلی استفاده خواهد شد. اگر شما این پیشنهاد را می پذیرید، باید Directive موسوم به ServerTokens را به مقدار ProductOnly تغییر دهید که نحوه آن در قبل ذکر شد.

دلیل این که پیشنهاد می شود که تنها یک ماکرو را تغییر دهید این است که برخی از ماژول ها(مثل mod_ssl) تنها با برخی از نسخه ها کار می کنند و از کار کردن با نسخه های دیگر خودداری می کنند.

یک راه دیگر برای تغییر نام این است که تابع ap_set_version() را تغییر دهید. این تابع مسئول نام گزاری سرور برای اولین بار است. برای مثال می توانید تابع موجود را (در http_main.c) با کدی شبیه به زیر تعویض کنید:

```
static void ap_set_version(void)
{
    /* set the server name */
    ap_add_version_component("Microsoft-IIS/5.0");
    /* do not allow other modules to add to it */
    version_locked++;
}
```

برای آپاچی ۲، تابع زیر را (در core.c) تعویض کنید:

```
static void ap_set_version(apr_pool_t *pconf)
{
    /* set the server name */
    ap_add_version_component(pconf, "Microsoft-IIS/5.0");
    /* do not allow other modules to add to it */
    version_locked++;
}
```

۲.۲. عوض کردن نام بوسیله mod_security

اگر نخواستید که در کد منبع دستکاری کنید، باید از ماژولهای طرف سومی استفاده کنید. مثلا mod_security. این ماژول باعث می شود که نام سرور تغییر کند. اما ابتدا باید اجازه بدهیم که سرور تمام اطلاعات خود را آشکار کند، سپس mod_security را تنظیم می کنیم تا هویت را عوض کند. این directive ها می تواند به فایل تنظیمات آپاچی اضافه شود:

```
# Reveal full identity (standard Apache directive)
ServerTokens Full
# Replace the server name (mod_security directive)
SecServerSignature "Microsoft-IIS/5.0"
```

Mod_security می تواند مکان حافظه ای را که نام سرور و مشخصاتش را نوشته شده است، تغییر دهد که این کار توانایی بالاتری را نسبت به تغییرات در کد منبع در اختیارش می گذارد. ضمنا مقدار ServerTokens باید Fullباشد.

۳. عوض کردن هویت سرور توسط mod_header در آپاچی ۲

این ماژول برای تغییر محتوای header پاسخ در آپاچی ۲ تهیه شده است. اما با وجود این شما نمی توانید دو header مهم را عوض کنید: Server و Date. البته این روش در زمانی که سرور در حالت Reverse Proxy کار می کند، مفید خواهد بود. در این صورت شما می توانید از تنظیمات زیر استفاده کنید:

```
Header set Server "Microsoft-IIS/5.0"
```

البته این راه مشکل جدی دارد چرا که هنگامی که یک منبع غیر واقعی از طرف مشتری درخواست می شود و سرور یک پاسخ خطای ۴۰۴ باز می گرداند، دیگر mod_header کار نمی کند و در واقع، هویت اصلی وب سرور را باز می گرداند.

۴. حذف محتوای پیش فرض

برای جلوگیری از شناخت آپاچی از روی فایل های پیش فرضی که در هنگام نصب در درون پوشه های مختلف قرار می گیرد، و همچنین عدم سوء استفاده از آنها، باید آن ها را پاک کرد. پس پوشه های زیر را پاک کنید:

- */usr/local/apache/cgi-bin*
- */usr/local/apache/htdocs*
- */usr/local/apache/manual* (Apache 2 only)

شاید نیاز باشد که پوشه /usr/local/apache/logs را نگه دارید، حتی با وجود این که شما ثبت وقایع را در /var/www/logs ثبت می کنید، چرا که برخی از ماژول ها ممکن است از این پوشه برای ذخیره فایلهای موقتی خود استفاده کنند.

تنها موردی که باقی می ماند این است که، نوع خطاهایی که به طرف کاربر باز میگردد را عوض کنیم. این صفحات می توانند بوسیله Directive موسوم به ErrorDocument پاک شوند.با استفاده از یک directive به ازای هر کد خطا، تمامی خطاهای http را عوض می کنیم:

```
ErrorDocument 401 /error/401.html
ErrorDocument 403 /error/403.html
ErrorDocument 404 /error/404.html
ErrorDocument 500 /error/500.html
...
```

یک راه حل برای این که تعداد زیادی صفحه برای کد خطاهای مختلف درست نکنیم این است که اسکریپت هوشمند بنویسیم که با گرفتن کد خطا، صفحه مناسب برای آن را تولید کند.