عنوان: **چک لیست امنیتی پروژه های نرم افزاری تحت وب**
نویسنده: ح مرادف
تاریخ: ۱۳۹۴/۰۵/۱۰ ۱۲:۵
آدرس: www.dotnettips.info
گروه‌ها: ASP.Net, MVC, Security

مقدمه:

امروزه یکی از بزرگترین دغدغه‌های فعالان حوزه آی تی، برقراری امنیت اطلاعات می‌باشد. با پدید آمدن بانک‌های داده‌ای آماری و مالی، حساسیت مسئله صد چندان می‌شود. در ادامه چک لیستی را ارائه می‌نمایم که با کمک آن می‌توانید تا حدود بسیار خوبی امنیت نرم افزار تحت وب خود را برقرار نمایید. در برخی از موارد مثال‌هایی از تکنولوژی مایکروسافت آورده شده است که این بدلیل تخصص نویسنده در تکنولوژی‌های مایکروسافت می‌باشد. در صورتیکه شما از تکنولوژی‌ها و زبان‌های سورس باز بهره می‌برید، می‌بایست معادل مورد ذکر شده را در زبان مورد استفاده خود بیابید .
ابتدا اجازه دهید مقداری با حملات آشنا شویم و سپس راه مقابله را در کنار هم بررسی نماییم.

مهمترین و خطرناک‌ترین حملات سطح وب :

حمله XSS

این نوع حملات بدین صورت است که هکر با استفاده از فرم‌های عمومی یا خصوصی (پنل‌های سایت) اقدام به ثبت کدهای مخرب جاوااسکریپت درون دیتابیس شما می‌نماید. همانطور که می‌دانید پایه اصلی سیستم‌های احراز هویت، ساخت فایل کوکی بر روی کامپیوتر کاربران می‌باشد. زمانی که مطلب ثبت شده‌ی هکر برای کاربران شما نمایش داده می‌شود، کدهای جاوا اسکریپت هکر روی مرورگر کاربر، اجرا شده و اطلاعات کوکی‌های کاربر به راحتی برای سایت هکر ارسال می‌شود (معمولا هکر یک صفحه روی وب می‌سازد تا بتواند اطلاعات دریافتی از کدهای جاوا اسکریپت خود را دریافت و در جایی ذخیره کند).
حال هکر به راحتی کوکی را بر روی مرورگر خودش تنظیم می‌کند و بعد وارد سایت شما می‌شود. سیستم شما او را با کاربر شما اشتباه می‌گیرد و به راحتی هکر به اطلاعات پنل کاربری کاربر(ان) شما دست پیدا می‌کند.

حمله SQL Injection

این حمله معروفترین حمله است که تقریبا با قدرت می‌توانم بگویم که درتکنولوژی ASP.Net با امکانات فوق العاده‌ای که بصورت توکار در دات نت در نظر گرفته شده است، بصورت کامل به فراموشی سپرده شده است. فقط 2 تا نکته‌ی ریز هست که باید در کدهایتان رعایت کنید و تمام.
این حمله بدین صورت است که هکر یک سری دستورات SQL را در کوئری استرینگ، به صفحات تزریق می‌کند و بدین صورت می‌تواند در کدهای کوئری TSQL شما اختلال ایجاد کند و اطلاعات جداول شما را بدست بیاورد. در این نوع حمله، هکر از طریق باگ سطح کد نویسی کدهای نرم افزار، به دیتابیس حمله می‌کند و اطلاعاتی مثل نام کاربری و کلمه‌ی عبور ادمین یا کاربران را می‌دزد و بعد می‌رود داخل پنل و خرابکاری می‌کند.

حمله CSRF

این حمله یکی از جالب‌ترین و جذاب‌ترین نوع حملات است که هوش بالای دوستان هکر را نشون می‌دهد. عبارت CSRF مخفف Cross Site Request Forgery است (احتمالا دوستان ام وی سی کار، این عبارت برایشان آشناست).
در این نوع حمله هکر یک فایل برای کاربر شما از طریق ایمیل یا روش‌های دیگر ارسال می‌کند و کاربر را به این سمت سوق می‌دهد که فایل را باز کند. کاربر یک فایل به ظاهر معمولی مثل عکس یا ... را می‌بیند و فایل را باز می‌کند. وقتی فایل باز می‌شود دیتای خاصی دیده نمی‌شود و گاهی هم اروری مبنی بر ناقص بودن فایل یا ... به کاربر نمایش داده می‌شود و کاربر فکر می‌کند که فایل، ناقص برای ارسال شده ...
اما در حقیقت با کلیک بر روی فایل و باز کردن آن یک درخواست POST از کامپیوتر کاربر برای سایت شما ارسال می‌شود و در صورتیکه کاربر در آن زمان در سایت شما لاگین باشد، سایت درخواست را با روی باز می‌پذیرد و درخواست را اجرا می‌کند. بدین

صورت هکر می‌تواند درخواست‌هایی را به سرویس‌های سایت شما که مثلا برای حذف یک سری داده است، ارسال کند و اطلاعات کاربر را حذف کند.

حمله Brute Force

در این حمله، هکر از یک سری برنامه برای ارسال درخواست‌های مکرر به فرم‌های سایت شما استفاده می‌کند و بدین صورت فرم‌های عمومی سایت شما مورد حجوم انبوهی از درخواست‌ها قرار می‌گیرد که این امر در بهترین حالت موجب ثبت کلی دیتای اسپم در دیتابیس شما و در بدترین حالت موجب داون شدن سایت شما می‌شود.

حمله DDOS

این نوع حمله مانند حمله Brute Force است؛ با این تفاوت که درخواست به همه‌ی صفحات شما ارسال می‌شود و معمولا درخواست‌ها از چندین سرور مختلف برای سایت شما ارسال می‌شوند و حجم درخواست‌ها به قدری زیاد است که عملا سرور شما هنگ می‌کند و کاملا از دسترس خارج می‌شود. این نوع حمله در سطح کد راه حل زیادی ندارد و در سطح سرور و فایروال باید حل شود و حل آن هم بدین صورت است که درخواست‌های بیش از حد طبیعی از یک آی پی خاص تشخیص داده شده و به سرعت، آی پی بلاک می‌شود و از آن به بعد درخواست‌های آن آی پی در فایروال از بین می‌رود و دیگه به سرور نمیرسد.

حمله SHELL

شل فایلی است خطرناک که اگر بر روی سرور سایت شما آپلود و اجرا شود، هکر از طریق آن دسترسی کاملی به کل سرور سایت شما خواهد داشت. فایل‌های دیگری با نام بک‌دور [1] نیز وجود دارند که نویسنده تمایل دارد آنها را نیز از نوع حمله SHELL معرفی نماید. این نوع از فایل‌ها به مراتب بسیار خطرناک‌تر از فایل‌های شل می‌باشند؛ تا جایی که ممکن است سال‌ها هکر به سروی دسترسی داشته باشد و مدیر سرور کاملا از آن بی خبر باشد. اینجاست که باید شدیدا مراقب فایل‌هایی که روی سایت شما آپلود می‌شوند باشید. نویسنده به تمامی خوانندگان پیشنهاد می‌نماید، در صورتیکه نرم افزار حساسی دارند، حتما از سرور اختصاصی استفاده نمایند؛ چرا که در هاست‌های اشتراکی که در آنها فضا و امکانات یک سرور بصورت اشتراکی در اختیار چندین سایت قرار می‌گیرد، وجود باگ امنیتی در سایر سایت‌های موجود بر روی سرور اشتراکی می‌تواند امنیت سایت شما را نیز به مخاطره بیاندازد. نویسنده تهیه‌ی سرور اختصاصی را شدیدا به توسعه دهندگان سایت‌های دارای تراکنش‌های بانکی بالا (داخلی یا خارجی) پیشنهاد می‌نماید. زیرا درگاه تراکنش‌های بانکی بر روی آی پی هاست شما قفل می‌شوند و در صورتیکه سرور بصورت اختصاصی تهیه شده باشد، آی پی سرور شما فقط و فقط در اختیار شماست و هکر نمی‌تواند با تهیه هاستی بر روی سرور اشتراکی شما، به راحتی آی پی قفل شده در درگاه بانکی شما را در اختیار داشته باشد. بدیهی است تنها در اختیار داشتن آی پی سرور شما جهت انجام خرابکاری در درگاه بانکی شما کافی نیست. ولی به نظر نویسنده این مورد در بدترین حالت ممکن 30% کار هکر می‌باشد. البته بحث حمله شل به سطح مهارت متخصصان سرورها نیز بستگی دارد. نویسنده اظهار می‌دارد اطلاعات دقیقی از تنظیماتی که بتواند جلوی اجرای انواع شل و یا جلوی دسترسی فایل‌های شل را بگیرد، ندارد. بنابراین از متخصصان این حوزه دعویت می‌نماید اطلاعاتی درباره این موضوع ارائه نمایند.

حمله SNIFF

در این نوع حملات، هکر پکت‌های رد و بدل شده‌ی بین کاربران و سرور شما را شنود می‌نماید و به راحتی می‌تواند اطلاعات مهمی مثل نام کاربری و رمز عبور کاربران شما را بدست آورد.

چک لیست امنیتی پروژه‌های نرم افزاری تحت وب

- بررسی کامل ورودی‌های دریافتی از فرم‌های سایت؛ هم در سمت کلاینت و هم در سطح سرور .
- در تکنولوژی دات نت به منظور تمیز سازی ورودی‌ها و حذف تگهای خطرناکی همچون تگ script، کتابخانه‌ای با نام Microsoft.Security.Application وجود دارد. کتابخانه‌های سورس باز دیگری نیز وجود دارند که نمونه آن کتابخانه AntiXss [2] سایت نوگت [3] می‌باشد.

- بررسی کامل ورودی‌های دریافتی از کوئری استرینگ‌های [4] سایت. اگر از ASP.Net MVC استفاده می‌نمایید، تا حدی زیادی نیاز به نگرانی نخواهد داشت، زیرا تبدیلات [5] در سیستم Model Binding انجام می‌پذیرد و این موضوع تا حد زیادی شما را در برابر حملات SQL Injection مقاوم می‌نماید.

- حتما در فرم‌های عمومی سایتتان از تصویر کپچا با امنیت بالا استفاده نمایید. این موضوع جهت شناخت روبات‌ها از انسان‌ها می‌باشد و شما را در برابر حملات Brute Force مقاوم می‌نماید.

- حتما سیستم شخصی سازی صفحات ارور را فعال نمایید و از نمایش صفحات ارور حاوی اطلاعات مهمی مانند صفحات ارور ASP.Net جلوگیری نمایید. این موضوع بسیار حساس می‌باشد و می‌تواند نقاط ضعف نرم افزار شما را برای هکر نمایان کند. حتی ممکن است اطلاعات حساسی مانند نام بانک اطلاعاتی، نام کاربری اتصال به بانک اطلاعاتی و نام جداول بانک اطلاعاتی شما را در اختیار هکر قرار دهد.

- استفاده از ORM ها یا استفاده از پروسیجرهای پارامتریک. این موضوع کاملا شما را در برابر حملات SQL Injection مقاوم می‌نماید. کما اینکه ORM ها، سطحی از کش را بصورت توکار دارا می‌باشند و این موضوع در سرعت دستیابی به داده‌ها نیز بسیار تاثیر گذار است. از طرف دیگر بانک اطلاعاتی SQL نیز امکانات توکاری جهت کش نمودن پرس و جوهای [6] پارامتریک دارد.

- لاگ کردن ارورهای سطح کد و سطح روتینگ [7] . یکی از مهمترین خصیصه‌های پروژه‌های با کیفیت، لاگ شدن خطاهای سطح کد می‌باشد. این امر شما را با نقاط حساس و ضعف‌های نرم افزار آگاه می‌سازد و به شما اجازه می‌دهد به سرعت در جهت رفع آنها اقدام نمایید. لاگ نمودن خطاهای سطح روتینگ شما را از فعالیت‌های هکرها جهت یافتن صفحات لاگین و صفحات مدیریتی پنل مدیریتی سایت اگاه می‌نماید، همچنین شما را از حملات SQL Injection نیز آگاه می‌نماید.

- جلوگیری از ایندکس شدن صفحات لاگین پنل مدیریت سایت در موتورهای جستجو. بخش مهمی از عملیات هکر ها، قرار دادن روبات‌های تشخیص رمز بر روی صفحات لاگین می‌باشد که به نوعی می‌توان این نوع حملات را در دسته حملات Brute Force قرار داد. موتورهای جستجو یکی از ابزارهای مهم هکرها می‌باشد. عملیات هایی مانند یافتن صفحات لاگین پنل مدیریتی یکی از کاربردهای موتورهای جستجو برای هکرها می‌باشد.

- لاگ کردن ورود و خروج افراد به همراه تاریخ، زمان، آی پی افراد و وضعیت لاگین. با کمک این موضوع شما می‌توانید ورود و خروج کاربران نرم افزار خود را کنترل نمایید و موارد غیر طبیعی و مشکوک را در سریعترین زمان مورد بررسی قرار دهید.

- استفاده از روال‌های استاندارد جهت بخش "فراموشی کلمه عبور". همیشه از استانداردهای نرم افزارهای بزرگ پیروی نمایید. بدیهی است استانداردهای استفاده شده در این نرم افزارها بارها و بارها تست شده و سپس بعنوان یک روال استاندارد در همه‌ی نرم افزارهای بزرگ بکار گرفته شده است. استاندارد جهانی بخش "فراموشی کلمه عبور" که در اغلب نرم افزارهای معروف جهان بکار گرفته شده است، عبارت است از دریافت آدرس ایمیل کاربر، احراز هویت ایمیل وارد شده، ارسال یک نامه‌ی الکترونیکی [8] حاوی نام کاربری و لینک تنظیم کلمه عبور جدید به ایمیل کاربر. بهتر است لینک ارسال شده به ایمیل کاربر بصورت یکبار مصرف باشد. کاربر پس از کلیک بر روی لینک تنظیم کلمه عبور جدید، وارد یکی از صفحات سایت شده و می‌تواند کلمه‌ی عبور جدیدی را برای خود ثبت نماید. در پایان، کاربر به صفحه‌ی ورود سایت هدایت شده و پیامی مبنی بر موفقیت آمیز بودن عملیات تغییر کلمه‌ی عبور به او نمایش داده می‌شود. البته روال ذکر شده حداقل رول استانداردی می‌باشد و می‌توان در کنار آن از روال‌های تکمیل کننده‌ای مانند پرسش‌های امنیتی و غیره نیز استفاده نمود.

- قراردادن امکاناتی جهت بلاک نمودن آی پی‌ها و غیر فعال نمودن حساب کاربری اعضای سایت. در نرم افزار باید این امکان وجود داشته باشد که آی پی هایی که بصورت غیر طبیعی در سایت فعالیت می‌نمایند و یا مکررا اقدام به ورود به پنل مدیریتی و پنل کاربران می‌نمایند را بلاک نماییم. همچنین در صورت تخلف کاربران باید بتوان حساب کاربری کاربر خاطی را مسدود نمود. این موضوع می‌تواند بسته به اندازه پروژه و یا سلیقه تیم توسعه بصورت خودکار، دستی و یا هر دو روش در نرم افزار در تعبیه شود.

- امن سازی سرویس‌های ای جکس و چک کردن ای جکس بودن درخواست ها. حتما جلوی اجرای سرویس‌های درون نرم افزاری از بیرون از نرم افزار را بگیرید. سرویس‌های ای جکس یکی از این نوع سرویس‌ها می‌باشند که در نرم افزارها جهت استفاده‌های داخلی در نظر گرفته می‌شوند. در این نوع سرویس‌ها حتما نوع درخواست را بررسی نمایید و از پاسخگویی سرویس‌ها به درخواست‌های غیر ای جکسی جلوگیری نمایید. در ASP.Net MVC این امر توسط متد Request.IsAjaxRequest انجام می‌پذیرد .

- محدود کردن سرویس‌های حساس به درخواست‌های POST. حتما از دسترسی به سرویس هایی از نوع Insert,Update و Delete از طریق فعل GET جلوگیری نمایید. در ASP.Net MVC این سرویس‌ها را به فعل POST محدود نموده و در ASP.Net Web API این سرویس‌ها را به افعال POST,PUT و DELETE محدود نمایید.

- عدم استفاده از آی دی در پنل‌های کاربران بالاخص در آدرس صفحات (کوئری استرینگ) و استفاده از کد غیر قابل پیش بینی مثل GUID به جای آن. حتی الامکان بررسی مالکیت داده‌ها در همه بخش‌های پنل‌های کاربری سایت را جهت محکم کاری بیشتر انجام دهید تا خدای نکرده کاربر با تغییر اطلاعات کوئری استرینگ صفحات نتوانند به داده‌های یک کاربر دیگه دسترسی داشته باشند.

- حتی الامکان پنل مدیران را از کاربران بصورت فیزیکی جدا نمایید. این مورد جهت جلوگیری از خطاهایی است که ممکن است

توسط توسعه دهنده در سطح سیستم مدیریت نقش رخ دهد و موجب دسترسی داشتن کاربران به بخش هایی از پنل مدیریتی شود.

- استفاده از الگوریتم‌های کدگذاری ترکیبی و کد کردن اطلاعات حساس قبل از ذخیره سازی در بانک اطلاعاتی. اطلاعات حساسی مانند کلمات عبور را حتما توسط چند الگوریتم کدگذاری، کدگذاری نمایید و سپس درون بانک اطلاعاتی ذخیره نمایید.

- تنظیمات حساس نرم افزار را درون فایل web.config قرار دهید و حتی الامکان آنها را نیز کدگذاری نمایید . بصورتی که اطلاعات قابلیت دیکد شدن را داشته باشند.

- ساخت پروژه بصورت چند لایه. این موضوع جهت جلوگیری از دستیابی هکر به ساختار لایه‌های پروژه‌های شما می‌باشد. به بیان دیگر اگر نهایتا هکر بتواند به اطلاعات FTP هاست شما دست یابد، استفاده از تکنولوژی چند لایه در بدترین حالت هکر را از دستیابی به اطاعات لایه‌های زیرین نرم افزار باز می‌دارد. البته این کار برای هکرها غیر ممکن نیست، اما بسیار سخت و زمان بر می‌باشد.

- اشتراک گذاری اینترفیس در سرویس‌های خارج برنامه ای و عدم اشتراک گذاری کلاس اصلی. این موضوع از دستیابی هکر به بدنه سرویس‌ها و پیاده سازی‌های آنها جلوگیری می‌نماید.

- استفاده از تکنیک‌های مقابله با CSRF در همه سرویس‌های POST. در ASP.NET MVC اتریبیوتی با نام AntiForgery جهت مقاوم سازی سرویس‌ها از حملات CSRF وجود دارد. مکانیزم بدین صورت است که در تمامی فرم‌های سایت یک کد منحصر به فرد تولید می‌گردد که همراه درخواست GET به کامپیوتر کاربر ارسال می‌شود و در هنگام ارسال درخواست POST به سرور، صحت کد مورد نظر بررسی شده و در صورت صحت، اجازه‌ی اجرای سرویس به درخواست داده می‌شود. بدین صورت وقتی کاربر سایت شما فایل آلوده‌ای را باز می‌نماید، در خواست ارسالی هکر که توسط فایل باز شده، به سرور سایت ما ارسال می‌گردد، فاقد کد منحصر به فرد بوده و از اجرای سرویس جلو گیری می‌شود.

- استفاده از سیستم‌های مدیریت نقش امن مانند IDENTITY در ASP.Net MVC و یا استفاده از امکانات توکار دات نت در سیستم‌های مدیریت نقش شخصی سازی شده [9] . بدیهی است امنیت این سیستم‌ها بارها و بارها تست شده است.

- بررسی فرمت و پسوند فایل‌های آپلود شده. توجه نمایید که بررسی پسوند فایل‌ها کافی نبوده و فرمت فایل‌ها نیز می‌بایست بررسی شود. حتی نویسنده پیشنهاد می‌نماید فایل‌ها را به نوع‌های مرتبطشان تبدیل [10] نمایید. در حوزه هک باید نمودن انواع ویروس، تروجان، شل و بک دور [11] به فایل‌های تصویری و متنی یک امر بسیار رایج است. بنابراین حساسیت زیادی روی این موضوع قرار دهید. نویسنده توصیه می‌نماید کتابخانه‌های کاملی برای این موضوع تدارک ببینید تا در تمامی پروژه‌ها نیاز به ایجاد مجدد آنها نداشته باشید و سعی نمایید در هر پروژه این کتابخانه‌ها را تکمیل‌تر و بهتر نمایید.

- تنظیم IIS جهت جلوگیری از اجرای فایل‌های اجرایی در مسیر آپلود فایل‌ها. شاید جمله بیان شده به نظر ترسناک و یا سخت برسد، اما این کار با نوشتن چند تگ ساده در فایل Web.Config به راحتی قابل انجام است و نیاز به هیچ نوع کدنویسی ندارد.

- آپلود فایل‌ها در پوشه App_Data و دسترسی به فایل‌ها از طریق سرویس‌های خود شما. پوشه App_Data پوشه‌ای امن است و دسترسی مستقیم از طریق آدرس بار مرورگر به فایل‌های درون آن توسط IIS داده نمی‌شود و افراد فقط از طریق سرویس‌های خود شما می‌توانند به فایل‌های داخل این پوشه دسترسی داشته باشند. بدین صورت در سرویس‌های خود می‌توانید با تبدیل نمودن [12] فایل‌ها به نوع خودشان (تصویر. پی دی اف یا ...) هکر را نا امید نمایید. این موضوع شما را در مقابل حملات SHELL مقاوم می‌نماید.

- استفاده از تکنیک‌های لاگین چند سطحی برای پنل ادمین. در این روش شما حتی با داشتن نام کاربری و کلمه‌ی عبور ادمین، قادر نخواهید بود وارد پنل ادمین شوید. نویسنده ابزار می‌دارد که این روش، یک روش ابداعی می‌باشد که از ترکیبی از احرا هویت ساده توسط نام کاربری و کلمه‌ی عبور به همراه تکنیک‌های احراز هویت ایمیل و موبایل مدیریت سایت می‌باشد.

- استفاده از SSL بسیار اهمیت دارد. بالاخص اگر نرم افزار شما Service Oriented باشد و نرم افزار شما سرویس هایی جهت اتصال به اپلیکیشن‌های خارجی مثل اپلیکیشن اندروید دارد. این مورد در صفحات لاگین نیز بسیار مهم است و موجب می‌شود نام کاربری و کلمه عبور کاربران شما بصورت هش شده بین کامپیوتر کاربر و سرور شما رد و بدل شود و عملا شنود پکت‌ها فایده ای برای هکر نخواهد داشت، زیرا داده‌ها توسط الگوریتم‌های امنیتی که بین سرور و مرورگر کاربران توافق می‌شود کدگذاری شده و سپس رد و بدل می‌شوند.

Back Door [1]

/https://www.nuget.org/packages/AntiXss [2]

www. Nuget.org [3]

Query String [4]

Casting [5]

Procedure [6]

Routing [7]

Email [8]

Custom Role Provider [9]

Cast [10]

Back Door [11]

Cast [12]

نظرات خوانندگان

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۴/۰۵/۱۰ ۲۲:۴۳

یک اکستنشن مناسب جهت محافظت از حملات Denial of Service و Brute-force. این روش هم در سال‌های گذشته توسط استک‌اورفلو استفاده می‌شده.

نویسنده: امیر هاشم زاده
تاریخ: ۱۳۹۴/۰۵/۱۰ ۲۳:۴۴

در روش استک اورفلو جهت نال نبودن کلید کش هر دو متغیر HTTP_X_FORWARDED_FOR و REMOTE_ADDR رو بررسی کنید.

نویسنده: غلامرضا ربال
تاریخ: ۱۳۹۴/۰۵/۱۱ ۰:۱۴

ظاهرا مشکلاتی هم دارد این روش .
این کتابخانه هم جالب بود

نویسنده: مصطفی حسن زاده
تاریخ: ۱۳۹۴/۰۵/۱۱ ۱۰:۱۷

در رابطه با **حمله SNIFF** میشه بیشتر توضیح بدی؟

نویسنده: ح مرداف
تاریخ: ۱۳۹۴/۰۵/۲۳ ۱۴:۲

با سلام،
در این روش هکر خودش رو توی مسیر ورود و خروج پکت‌های سرور قرار میده و با کمک ابزارهایی پکت‌های ورودی و خروجی سرور رو شنود می‌کنه و اطلاعات رو ذخیره می‌کنه.
در این موارد اطلاعات حساسی مثل نام کاربری و کلمه عبور کاربران ک هدر صفحات ورود و یا حتی ثبت نام که بصورت Clear Text به سرور ارسال میشن، به راحتی در اختیار هکر قرار می‌گیرن.
بهترین راه حل مقابله با این قضیه استفاده از SSL روی سایت است. بدین صورت وقتی کاربر وارد یکی از صفحات سایت شما میشه، سرور یک پروتوکل امنیتی رو با مرورگر کاربر به اشتراک می‌گذاره و مرورگر کاربر متوجه میشه که باید بر اساس پروتکلی که دریافت کرده داده‌های ارسالیش رو در سمت کلاینت هش کنه و بعد برای سرور ارسال کنه. این روش امنیت بسیار بالایی داره.