ترجمه فا
TarjomeFa.Com

ارائه شده توسط:

سایت ترجمه فا

مرجع جدیدترین مقالات ترجمه شده

از نشریات معتبر

فیلتر ابر: کنترل عملی انتقال اطلاعات حساس به ابر

3-طراحی فیلتر ابر

هدف فیلتر ابر ارایه یک راه حل آسان و کاربردی برای کنترل انتشار داده ها بین یک شرکت و سرویس های ابر است. ما بر ارایه دهندگان فضای ذخیره ابری تاکید می کنیم که بیانگر داده های ذخیره شده نظیر فایل ها می باشند. یک لازمه مهم برای سیستم، قابلیت کاربرد آسان سیستم در ارایه دهندگان فضای ذخیره ابری مختلف با حداقل پیکر بندی است. در عین حال، بایستی قادر به سازگاری با API سرویس های فضای ابری ویژه می باشد. برای مثال، فیلتر ابر بایستی قادر به اجرای خطی مشی های انتشار داده هایی باشد که مانع از بارگذاری فایل ها توسط کاربران به پوشه های خاص می شوند: این اطلاعات در روش های HTTP وجود دارند.

نیاز به پشتیبانی از ارایه دهندگان فضای ذخیره ابری موجود، موجب شده است تا یک سری از قوانین و ملزومات در خصوص شیوه انتشار برچسب ها با داده ها ایجاد شوند.اولا، برچسب ها بایستی در ارتباط با فایل های ذخیره شده توسط ارایه دهنده فضای ذخیره ابری باشند. بسیاری از سرویس های ابر، امکان عملیات فایل از راه دور نظیر کپی، جا به جایی و دسترسی به نسخه های قبلی را می دهند. این عملیات بایستی برچسب های مربوط به فایل ها را مستقل از شیوه اجرای عملیات حفظ کنند. این مستلزم یک طرحی است که در آن برچسب ها درون فایل ها جاسازی می شوند. دوما، در صورتی که یک فایل در یک دستگاه کاربر دانلود شود، برچسب ها بایستی در ارتباط با فایل ها باقی بمانند.برچسب های ذخیره شده محلی نشان می دهند که چگونه فایل توسط ابر فیلتر در گذشته مدیریت شده است و اگر فایل دوباره بارگذاری شود، این اطلاعات را می توان برای کاهش نیاز به ورودی کاربر مورد استفاده قرار داد. به منظور بیشینه سازی پتانسیل پذیرش، مکانیسم ذخیره برچسب ها بایستی پلتفرم آگنوستیک[1] باشد. این مانع از استفاده از ویژگی های مختص سیستم های فایل(برای مثال مشخصه های فایل توسعه یافته در لینوکس یا جریان جایگزین داده

[1] platform-agnostic

در NTFS) می شود که نیازمند ترجمه برچسب ها بین الگو های مختلف بوده و ممکن است توسط سرویس های ذخیره ابری پشتیبانی نشود. ما یک رویکرد جا سازی برچسب ها در فایل ها را در بخش 3.2 توصیف می کنیم.

فیلتر ابر متشکل از اجزای زیر است(شکل 1):

پروکسی کلاینت وسرویس: دو پروکسی فیلتر ابر، کلاینت و سرویس، ترافیک HTTP بین سازمان و ارایه دهنده ابر را قطع می کند. هر پروکسی، داده های در حال انتقال را از طرف دامینی که بخشی از آن است بازرسی می کند. به علاوه، پروکسی سرویس می تواند خطی مشی تعیین شده توسط کلاینت های ارایه دهنده را اجرا کند. پروکسی ها مسئول برچسب زنی داده ها می باشند.

حافظه خط مشی: هر پروکسی یک حافظه خط مشی را دارد. حافظه خط مشی،مجموعه ای از قوانین رویداد-شرایط –عملکرد را انباشته می کند.

قوانین ECA، وقتی که انتقال فایل اتفاق می افتد، خط مشی انتشار داده ها را تعیین می کنند و از این روی، فعالیت های پروکسی ها را کنترل می کنند. درک فرمت ECA آسان بوده و موجب تسهیل پیاده سازی آن هنگام برقراری ارتباط با پروکسی می شود(رجوع شود به 3.1.3).

افزونه های مرور گر: یک افزونه مرور گر، اطلاعات مورد استفاده برای برچسب زنی فایل های بار گذاری شده را جمع آوری می کند. این افزونه با پروکسی کلاینت ارتباط برقرار کرده و می تواند صریحا کاربر را ترغیب به تایید آن در زمان شناسایی یک بارگذاری فایل توسط پروکسی کند.

شکل 1 نشان می دهد که چگونه آپلود فایل بین یک سازمان و یک ارایه دهنده ابر اتفاق می افتد. در گام 1، کاربر، فایل را از طریق یک فرم وب تحویل می دهد. پلاگین مرور گر، به درخواست HTTP خروجی، مجموعه ای از اطلاعات شناسایی را برای کاربر فعلی همراه با متادیتا در مورد فایل نظیر محلی که فایل به طور محلی ذخیره شده است الحاق می کند. در گام دوم،پروکسی کلاینت موجب قطع درخواست شده،

محتوی درون آن را بازرسی می کند و اطلاعات شناسایی کاربر را بازیابی می کند. سپس درخواست را با قواعد ECA در حافظه خط مشی آن مطابقت می دهد.

وقتی که قاعده ECA مطابق با یک درخواست باشد، پروکسی، یک عمل را اجرا می کند. در گام سوم، در مورد محرمانگی فایل در حال بارگذاری از کاربر پرس و جو می کند. پاسخ کاربر منجر به برچسبی می شود که به فایل الحاق شده است. بعد از ثبت درخواست برای حسابرسی آینده، به سرویس ابر هدایت می شود.

در گام پنجم، پروکسی سرویس، درخواست ها را همراه با برچسب های آن ها بازرسی می کند. حافظه خط مشی محلی، خط مشی بیان شده به صورت قواعد ECA را ذخیره می کند که ارایه دهنده ابر این قواعد را بر روی داده های کلاینت اجرا می کند. برای مثال، یک ارایه دهنده ابر می تواند از میزبانی فایل های حساس مربوط به یک سازمان خاص برای اجتناب از مشکلات حقوقی آینده، انصراف دهد. در صورتی که بارگذاری فایل پذیرفته شود، درخواست به یک سرویس ذخیره ای ارسال می شود.

در گام پنجم، یک درخواست HTTP بعدی، فایل بارگذاری شده در مراحل 1-4 را بازیابی می کند. پروکسی سرویس، پاسخ سرویس ذخیره سازی را را قطع کرده و از برچسب متصل به داده ها برای تصمیم گیری در مورد شیوه پاسخ به درخواست استفاده می کند. پروکسی سرویس می تواند با پروکسی کلاینت برای بدست آوردن یک خط مشی برای اجرا از طرف آن تماس برقرار کند برای مثال برای اجتناب از آزاد سازی داده ها به یک کاربری که در شبکه سازمانی نیست.

3.1 تعیین خط مشی انتشار داده ها

فیلتر ابر، انتشار داده ها را با استفاده از برچسب ها و قواعد رویداد-شرایط-عملکرد(ECA) کنترل می کند. قواعد ECA، خط مشی های مختلف را رمز گذاری کرده و برچسب های متصل به فایل ها تعیین می کنند که کدام خط مشی خاص را در زمان قطع یک فایل در حال ارسال، اجرا کنند.

3.1.1 برچسب ها

فیلتر ابر، برچسب ها را به فایل های در حال ارسال بین دامین ها ضمیمه می کند. چون بر چسب ها در زمان انتشار بین سازمان ها متصل به فایل ها هستند، برچسب ها به عنوان یک ضمیمه قابل اطمینان فایل ها با خط مشی های کنترل کننده انتشار آن ها عمل می کنند. هم چنین برچسب ها، متادیتا های اضافی مربوط به اجرای یک خط مشی خاص را ذخیره می کنند. چون برچسب ها همه متادیتا های لازم برای اجرای خط مشی را برچسب گذاری می کنند، پروکسی های فیلتر ابر قادر به اجرای خط مشی های طراحی شده تنها با بازرسی برچسب ها بوده و امکان یک طراحی پروکسی بدون حالت ساده را می دهند.

هر برچسب متشکل از سه بخش است1- یک شناسه 2- یک مجموعه ای از پارامتر های نام گذاری شده و 3-آدرس پروکسی فیلتر ابر معتبر برای آن برچسب. شناسه یک نام متنی و کاربر پسند از خط مشی انتشار داده های خاص است. برای مثال، شناسه کاربرخصوصی، می تواند توسط یک سازمان در اشاره به خط مشی ای به کار رود که تسهیم فایل ها در یک سرویس ذخیره ابر را محدود می کند. پارامتر های نام گذاری شده در یک برچسب، امکان سفارشی سازی خط مشی انتشار داده ها را با متادیتای دلخواه می دهد. برای مثال، پارامتر کاربر=[ip108, prp] برای خط مشی کاربر-خصوصی، تعیین کننده این است ک یک فایل را تنها می توان بین یک فهرست ویژه از کاربران به اشتراک گذاشت. همان طور که در بخش 3.1.3 توضیح داده شد، یک آدرس پروکسی معتبر موجب حصول اطمینان از اتصال منحصر به فرد برچسب با یک خط مشی انتشار داده می شود.

3-1-2 قواعد ECA

هر قاعده ECA زمانی اجرا می شود که یک رویداد اتفاق بیفتد. در صورتی که شرط مربوط به یک رویداد برقرار باشد، یک عمل اجرا می شود.

رویداد ها

چون فیلتر ابر به طور ویژه سرویس های ذخیره ابر را هدف یابی می کند که در آن HTTP یک پروتوکل انتقال غالب است، رویدادی که موجب تحریک فعال سازی قاعده ECA می شود، موسوم به روش درخواست

HTTP است. یک قاعده ممکن است برای درخواست های ورودی HTTP که از خارج از دامین هستند و یا برای درخواست های خروجی HTTP از درون دامین تحریک شود. یک مدیر می تواند رویداد های تحریک شده با درخواست روش HTPP خاص و یا برای درخواست های ارسالی به URI های خاص را تعیین کند.

برای مثال، رویداد زیر را در نظر بگیرید:

$$e_{uploads}: \{out\}\ \{put\ post\}\ \{\texttt{(.*\.)*dropbox.com(/.*)*}\}$$

این رویداد مطابق با همه درخواست های PUT و POST خروجی به سمت سرور های دراپ باکس است. این از عبارت با قاعده برای انطباق درخواست HTTP URI استفاده می کند. یک سازمان می تواند از این قاعده برای پیش گیری از آپلود همه فایل ها به دراپ باکس استفاده کند.

شروط

فیلتر ابر، از برچسب ها به عنوان پیش شرط های لازم برای تحریک اعمال استفاده می کند. هر شرط با وجود یک فایل برچسب زنی شده در یک درخواست یا پاسخ HTTP برقرار می شود.

شروط می توانند به صورت سرویس-آگنوستیک یا سرویس- ویژه باشند. برای شروط سرویس-آگنوستیک، HTTP API سرویس نادیده گرفته می شود. این شروط با وجود هر فایل برچسب زنی شده در یک درخواست یا پاسخ HTTP برقرار می شوند. در عوض، شروط سرویس ویژه مستلزم وجود یک برچسب در یک پارامتر ویژه یا بخشی از یک پاسخ/درخواست می باشد. چون پاسخ ها و درخواست های HTTP قادر به ذخیره فایل های مختلف در پارامتر های مختلف میباشند و هر فایل را می توان با یک برچسب متفاوت برچسب زنی کرد، یک شرط سرویس ویژه تعیین کننده مجموعه ای از پارامتر های HTTP و برچسب هایی است که بایستی متصل به مقدار هر پارامتر باشد.

به عنوان یک مثال، شرط سرویس ویژه را در نظر بگیرید:

$$c_{bank}: file \Rightarrow \{secret\}\ \{\texttt{cf\.bank\.com}\ \}$$

این شرط مطابق با همه درخواست های HTTP است که حاوی یک پارامتر موسوم به فایل با یک مقدار برچسب Secret تعریف شده با پروکسی فیلتر ابر در cf.bank.com می باشد(عبارت با قاعده). این شرط می تواند توسط یک پروکسی سرویس برای پیش گیری از آپلود فایل های حاوی داده های محرمانه استفاده شود.

عملیات

یک عمل، تعیین کننده خط مشی انتشار داده های یک سازمان یا سرویس ابر با توجه به کلاس داده هایی است که قاعده ویژه را فعال می کند. عملیات، اسکریپت هایی هستند که از API فیلتر ابر ویژه نشان داده شده در جدول 1 استفاده می کنند و توسط یک پروکسی اجرا می شوند.

سه عملیات پایه برای اسکریپت ها، شامل issue، return و log می باشند. این عملیات به ترتیب برای ایجاد، پاسخ به و ذخیره درخواست های HTTP برای حسابرسی آینده استفاده می شوند. یک اکشن اسکریپتیاتی می تواند از این روش ها برای دست یابی به معنای allow/deny فایروال های سطح شبکه استفاده کنند. سه روش بعدی،getLabels attLabelوdetLabel ، برچسب های فایل های در حال انتقال را دستکاری و اصلاح می کنند. برچسب ها، تنها مکانیسم موجود برای اسکریپت ها برای ذخیره داده ها در درخواست های مختلف HTTP می باشند- اسکریپت ها به خودی خود بدون حالت هستند. تصمیم برای ضمیمه یک برچسب به یک فایل بستگی به داده های واقعی در حال ارسال دارد که از طریق getContent و یا بر روی ورودی کاربر با استفاده از روش Ask قابل دسترسی است(بخش 3.2).

3.1.3 خط مشی انتشار داده های توزیعی

یک پروکسی می تواند قواعد ECA را از طرف پروکسی دیگر بدست آورده و اجرا کند. این وضعیت زمانی رخ می دهد که یک سازمان بخواهد تا داده ها را در یک ابر ذخیره کندو این موجب تحمیل محدودیت هایی بر شیوه دسترسی به داده ها می شود. در این سناریو ها، پروکسی کلاینت باید به پروکسی سرویس اعتماد

کند و قواعد ECA را در اختیار او قرار دهد در حالی که پروکسی سرویس بایستی به پروکسی کلاینت به عنوان منبع قواعد ECA اعتماد کند.

انتشار قاعده ECA بین پروکسی ها زمانی رخ می دهد که یک پروکسی، یک درخواست HTTP را قطع می کند. دو پیش شرط باید برقرار باشد:1- درخواست HTTP بایستی دارای یک برچسبی باشد که حاوی آدرس معتبر متفاوتی از آدرس پروکسی فعلی است 2- پروکسی فاقد قواعد ECA با شروط قابل ارجاع به این برچسب است. سپس پروکسی مستقیما با پروکسی معتبر برای دریافت مجموعه ای از قواعد ECA که بایستی آن ها را اجرا کند تماس برقرار می کند. این قواعد ECA خارجی تنها به برچسب های شرایطی که دارای آدرس معتبر یکسان همانند مبدا آن هاست استناد می کنند. به این ترتیب اطمینان حاصل می شود که قواعد ECA خارجی نمی تواند با اجرای سایر خط مشی های انتشار داده ها در یک پروکسی تداخل ایجاد کند.

به علاوه، قواعد ECA خارجی یک پارامتر دامین را تعیین می کنند که محدود کننده پروکسی های دور دست و واجد شرایط بوده و یک پارامتر timeout را برای اطمینان از جدید بودن تعین می کند. این قواعد به طور محلی استفاده نمی شوند بلکه در صورت درخواست و تقاضا به پروکسی های دور دست منتشر می شوند. چون انتشار قاعده ECA بعد از تشخیص انتقال فایل واقعی رخ می دهد، خط مشی های جدید لزوما به طور صریح استفاده نخواهند شد.

3-2 اجرای خط مشی انتشار داده ها

برای این که فیلتر ابر کاربردی و عملی باشد، ما بایستی نیاز به ورودی کاربر در زمان الحاق برچسب ها به فایل ها را به حداقل برسانیم و شرایط انتشار موثر برچسب ها را در دامین ها فراهم کنیم. ما یک نمونه سیستم اولیه را توسعه داده ایم که در آن پروکسی در فیتون اجرا شده و افزونه مرور گر یک افزونه جاوا اسکریپت برای فایرفاکس است. ما در مورد این مسائل در نمونه خود بحث می کنیم.

3-2-1 نمونه سازی برچسب

اسکریپت عمل، زمانی که انتقال فایل، یک قاعده ECA را فعال سازی می کند،برچسب ها را به فایل ها ضمیمه می کند. چهار روش مختلف را می توان برای استنباط برچسب صحیح برای فایل در حال انتقال مورد استفاده قرار دارد. اول، اکشن اسکریپت ها می توانند به خود فایل دسترسی پیدا کنند. این امکان برچسب زنی را بر اساس استفاده از محتوی خاص در فایل برای طبقه بندی امنیت می دهد برای مثال، استفاده از کلمه: طبقه بندی شده"Classified" در اسناد.

دوما، افزونه مرور گر، اطلاعات زمینه ای را از دستگاه کاربر به درخواست های HTTP متصل می کند. امروزه، این شامل ورود کاربر، محل فایل در سیستم فایل و ویژگی های آن است. یک اکشن اسکریپت می تواند از این اطلاعات برای استنباط محرمانگی برای فایل های ذخیره شده در یک محل شبکه ویژه استفاده کند.

سوما، فایل های دانلود شده از یک سرویس ذخیره سازی ابر دارای برچسب هایی ازتعاملات قبلی با فیلتر ابر می باشند.این برچسب ها در صورتی مفید هستند که فایل مجددا بارگذاری شود: آن ها را می توان برای اجتناب از پرس و جوی کاربر استفاده کرد. برای اطمینان از عدم تصمیم گیری بر اساس برچسب هایی که دیگر معتبر نیستند، می توان فایل ها را درون برچسب ها ذخیره کرد و تغییرات فایل را شناسایی کرد.

در نهایت، پروکسی کلاینت می تواند از کاربر در طی بار گذاری درخواست ورودی کند(یعنی با استفاده از روش Ask در جدول 1). افزونه مرور گر، یک فرم HTTP تولید شده دینامیکی را با اکشن اسکریپت دریافت کرده، و آن را به کاربر نشان داده و پاسخ کاربر را به پروکسی انتقال می دهد. محتوی فرم بستگی به سیاست اجرا شده دارد.

3-2-2 انتشار برچسب

به منظور جاسازی برچسب های فیلتر ابر، می توان از مفهوم متادیتا استفاده کرد که بسیاری از فرمت های فایل را پشتیبانی می کند. در نمونه ارایه شده، ما از پلتفرم متادیتا قابل توسعه Adobe(XMP)(1) استفاده می کنیم.XMP مشخصه ای برای نشان دادن فرا داده دلخواه در RDF/XML بوده و آن را درون فرمت های مختلف فایل ذخیره می کند. یک SDK برای جا سازی متادیتا XMP به صورت برنامه نویسی

در فرمت های فایل مختلف وجود دارد(PDF، EPS و JPEG). شکل 2 نشان می دهد که چگونه برچسب کاربر- خصوصی از بخش 3.1.1 را می توان در XMP نشان داد.

ما هم چنین از XMP برای ذخیره برچسب های درون اسناد مایکروسافت افیس استفاده می کنیم. فرمت های فایل افیس بعد از نسخه 2007، از قرار داد های بسته بندی باز(7) پیروی می کنند. هر بسته(برای مثال یک فایل ورد) یک آرشیو زیپ از هر دو فایل های متنی(برای مثال XML) و دو دویی(برای مثال تصاویر) می باشد که موسوم به بخش یا Part است. ویژگی های برنامه- ویژه نظیر تعداد کاراکتر های کل در یک سند ورد یا برچسب های فیلتر ابر رمزگذاری شده XMP را می توان در بخش های اضافی درون بسته ذخیره کرد. چون برنامه های افیس، بخش هایی از انواع اسناد ناشناخته را نادیده می گیرند، مسائل سازگاری در برچسب های XMP وجود ندارند.

ترجمه فا
TarjomeFa.Com

این مقاله، از سری مقالات ترجمه شده رایگان سایت ترجمه فا میباشد که با فرمت PDF در اختیار شما عزیزان قرار گرفته است. در صورت تمایل میتوانید با کلیک بر روی دکمه های زیر از سایر مقالات نیز استفاده نمایید:

✔ لیست مقالات ترجمه شده

✔ لیست مقالات ترجمه شده رایگان

✔ لیست جدیدترین مقالات انگلیسی ISI

سایت ترجمه فا ؛ مرجع جدیدترین مقالات ترجمه شده از نشریات معتبر خارجی