باسمه تعالی

عنوان مستند

بررسی آسیب پذیری منع سرویس در سیستم فایل NTFS ویندوز

فهرست مطالب

۱ چکیده

سیستم فایل الگویی است که برای ذخیره، بازیابی و سازماندهی فایل‌ها و داده‌ها بر روی حافظه‌ها مورد استفاده قرار می‌گیرد یعنی داده‌های مشخص توسط سیستم فایل به بخش یا بخش‌های مجزایی تبدیل می‌شوند که این بخش‌ها درواقع همان فایل‌ها هستند. یکی از انواع سیستم فایل، NTFS[۱] نام دارد که سیستم فایل مورد استفاده در ویندوزهای خانواده NT به بعد است.

اخیرا یک آسیب‌پذیری denial of service در سیستم فایل NTFS گزارش شده است. مهاجم با استفاده از این باگ در ویندوز و با فریب دادن کاربر می‌تواند فایل غیر موجود با یک مسیر نادرست را باز کرده و از طریق فراخوانی ساده‌ی یک فایل شامل $MFT[۲] مربوط به ویندوز، می‌تواند منجر به crash کردن ویندوز گردد. درنتیجه با استفاده از این آسیب‌پذیری و دسترسی از راه دور به یک سیستم می‌توان عملیات آن را مختل نمود.

۲ محصولات تحت تاثیر

اکثر نسخه‌های ویندوز دارای این آسیب‌پذیری هستند شامل: windows Vista, Windows۷, Windows۸, Windows۸,۱. اما Windows۱۰ فاقد این آسیب‌پذیری است.

۳ تاثیر آسیب پذیری

مهاجم با استفاده از این آسیب‌پذیری و دسترسی از راه دور به یک سیستم می‌تواند عملیات آن را مختل نماید. البته این مشکل نمی‌تواند در Chrome ایجاد شود زیرا مرورگر گوگل اجازه نمی‌دهد بارگیری تصاویر با مسیرهای نادرست مانند $MFT انجام شود. اما مرورگرهای Internet Explorer و Firefox اجازه‌ی این کار را می‌دهند و در نتیجه اگر قربانی از این مرورگرها استفاده کرده باشد سیستم crash خواهد کرد.

[۱] New Technology File System

[۲] Master File Table

۴ سابقه

این آسیب‌پذیری NTFS بسیار شبیه به آسیب پذیری دیگری است که در دهه ۱۹۹۰ کشف شد و به این صورت بود که سیستم به دلیل آسیب‌پذیری مسیر فایل "C: / con / con" از کار می‌افتاد. این آسیب‌پذیری در سیستم های ویندوز ۹۵ و ویندوز ۹۸ وجود داشت. وقتی کاربر در قسمت Run مسیر "C: / con / con" را وارد می‌کرد و تلاش می‌کرد تا به آن دسترسی یابد، سیستم crash می‌کرد.

۵ مشخصه های آسیب پذیری

در این قسمت به مرور کامل آسیب‌پذیری ذکر شده می‌پردازیم.

۵-۱ خلاصه‌ای از نحوه عملکرد سیستم فایل NTFS

قبل از توصیف ماهیت مشکل، بهتر است اصول اولیه ساخت سیستم فایل را مورد بررسی قرار دهیم. هنگامی که فرآیند مشخصی فایلی را باز می‌کند، به جز HANDLE دریافت شده در آن، در فضای هسته[۳]، ساختارها هم توسط خود هسته و هم توسط سیستم فایل تشکیل می شوند که در واقع، حجم فایل را در حافظه نشان می دهند. شکل(۱) این ساختارها را نشان داده شده است. فایل HANDLE همیشه به ساختار هسته FILE_OBJECT اشاره دارد. این ساختار قبل از ارسال درخواست به سیستم فایل، توسط هسته تولید می‌شود. سیستم فایل، به ترتیب فیلدهای این ساختار را مقداردهی می‌کند. بنابراین ساختار FILE_OBJECT حاوی اشاره‌گرهای زیر به ساختارهای سیستم فایل خواهد بود:

۱- FCB[۴] : بلوک کنترل فایل که حاوی تمام اطلاعات لازم برای مدیریت فایل است.

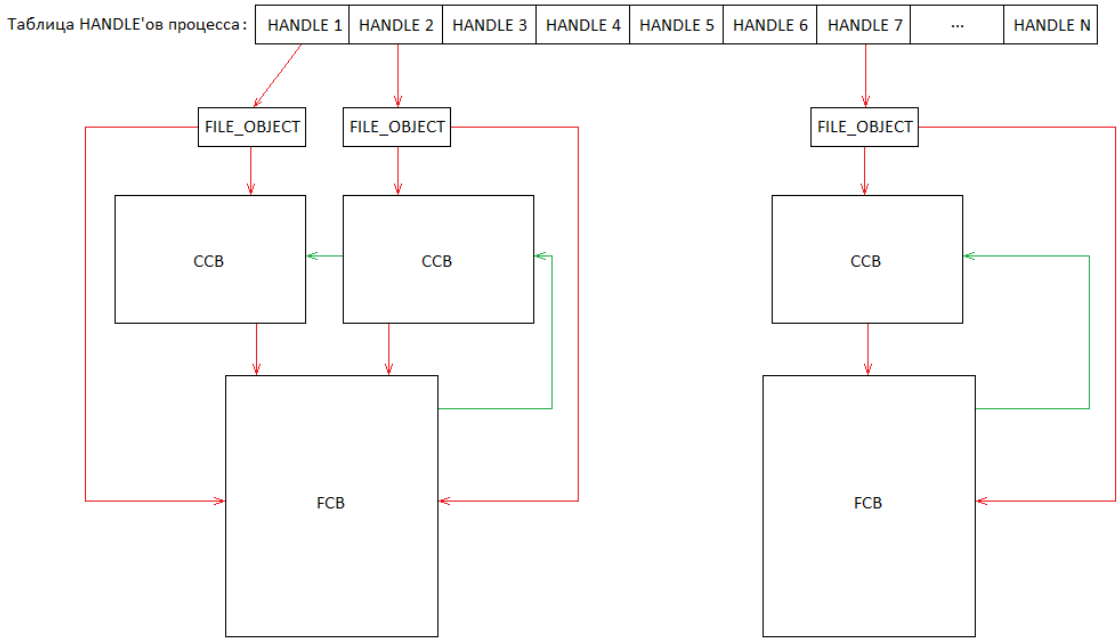
۲- CCB[۵] : یک ساختار سیستم فایل که داده‌های مختص یک نمونه از یک فایل باز شده را نگه می‌دارد.

---

[۳] Kernel space

[۴] File control block

[۵] Context Control Block



شکل(۱): جدول HANDLE

همچنین ممکن است دو HANDLE مختلف به یک فایل[۶] اشاره کنند که در سمت چپ شکل(۱) مشاهده می‌شود. ساختار FCB حاوی فهرستی از تمام ساختارهای CCB است. ساختار CCB حاوی اشاره‌گری به FCB مربوطه است. یعنی برای هر فایل باز در حافظه دقیقا یک ساختار FCB وجود خواهد داشت. اگر آن فایل چندین بار باز شده باشد دقیقا به همان تعداد ساختار CCB مربوط به آن فایل باز خواهد شد و همه‌ی این ساختارهای CCB به همان یک ساختار FCB اشاره خواهند کرد.

از آنجایی که فایل می‌تواند به طور همزمان توسط فرایندهای مختلف مورد دستیابی قرار گیرد، این عملیات‌های موازی باید سریال شوند. بعضی از عملیات می‌توانند به صورت همزمان انجام گیرند (مثل خواندن). اما شرایطی وجود دارد که دسترسی باید به صورت انحصاری باشد (مثل نوشتن یک رکورد). برای این منظور، هسته یک مکانیزم سریال‌سازی ERESOURCE را فراهم می‌کند. شیء[۷] ERESOURCE می‌تواند به صورت انحصاری یا اشتراکی نگهداری شود. اگر این شیء به صورت اشتراکی نگهداری شود، همه‌ی تلاش‌ها برای دستیابی به آن باید فورا پاسخ داده شوند اما اگر این شیء به صورت انحصاری نگهداری

[۶] Volume File
[۷] Object

شود و تلاشی برای تصاحب آن شیء آمده باشد و صف انتظار خالی نباشد، هر تلاش باید در صف انتظار قرار گیرد.

ساختار FCB از سیستم‌های فایل شامل این میکانیزم برای دسترسی سریال می‌باشند و همیشه از آن در طی دسترسی فایل استفاده می‌کنند. این مکانیزم یکپارچگی در حافظه را تضمین می‌کند.

فایل $mft از سیستم فایل NTFS، یک فایل سیستمی است. این فایل مکان تمام فایل‌ها را در حجم[۸] مربوطه مشخص می‌کند و بنابراین فایل بسیار مهمی است. NTFS در هنگام استقرار فایل‌ها $mft را برای استفاده شخصی خود باز می‌کند. هرگاه یک فایل می‌خواهد باز شود و یا محتوی یک دایرکتوری می‌خواهد خوانده شود، NTFS فایل $mft را می‌خواند. همچنین هر بار که یک فایل NTFS می‌خواهد ایجاد یا پاک شود، NTFS در این فایل می‌نویسد. بنابراین قبل از هر کدام از این عملیات‌ها روی یک فایل، شیء ERESOURCE از آن فایل تصرف خواهد شد و سپس آن عملیات اجرا شده و پس از آن، آن شیء آزاد خواهد شد.

۵-۲ مروری بر آسیب پذیری منع سرویس

آسیب‌پذیری منع سرویس یا به اختصار DoS نوعی از آسیب‌پذیری است که منجر به منع سرویس شده بدین معنی که مهاجم با استفاده از این آسیب‌پذیری تلاش می‌کند ماشین و منابع شبکه از دسترس کاربران مجازش خارج شود و منجر به قطع موقت یا دائمی و یا تعلیق خدمات یک میزبان متصل به اینترنت می‌شود. حمله DoS کامپیوتر هدف را وادار به ریست شدن یا از کار افتادن می‌کند، بنابراین نمی‌تواند به درخواست‌های مورد نظرش سرویس بدهد.

۵-۳ جزییات آسیب پذیری

برای درک ماهیت مشکلی که منجر به این آسیب‌پذیری شده، لازم است قاعده کلی تابع NtfsCommonCreate از سیستم فایل NTFS را بدانیم. شکل(۲) قسمت‌هایی از این تابع را نشان می‌دهد که به طور مستقیم مربوط به این مشکل است.

[۸] Volume

```
for ( ;; ) {

    NtfsFindStartingNode( &FullName, &RemainingName, &CurrentFcb );

    if ( RemainingName.Length == 0 ) {
        break;
    }

    FirstPass = true;

    for ( ;; ) {

        if ( !IsDirectory( &CurrentFcb ) ) {
            Status = STATUS_OBJECT_PATH_NOT_FOUND;
            break;
        }

        FsRtlDissectName( RemainingName, &Name, &RemainingName );

        if ( !FirstPass ) {
            NtfsReleaseFcbWithPaging( ParentFcb );
        }

        if ( RemainingName.Length == 0 ) {
            break;
        }

        ParentFcb = CurrentFcb;
        NtfsOpenSubdirectory( &Name, &CurrentFcb );

        FirstPass = false;

    }

    break;

}

if ( !NT_SUCCESS( Status ) ) {

    if ( CurrentFcb != NULL ) {
        NtfsTeardownStructures( CurrentFcb );
    }

}

return Status;
```

شکل(۲): قسمت‌هایی از تابع NtfsCommonCreate

سیستم فایل NTFS، درختی از فایل‌ها و یا دایرکتوری‌های باز شده را ذخیره می‌کند. بنابراین برای بهبود کارایی، پیدا کردن فایل هدف در این درخت نسبت به خواندن مکرر از آن حجم مربوطه مقرون به صرفه‌تر است. در نتیجه تابع NtfsCommonCreate سعی می‌کند با استفاده از تابع NtfsFindStartingNode، آن را پیدا کند. اگر فایل پیدا نشد، این تابع سعی می‌کند دایرکتوریی را که فایل هدف در آن قرار دارد پیدا کند. این تلاش تا رسیدن به ریشه‌ی آن سیستم فایل ادامه خواهد داشت. تابع NtfsFindStartingNode یک اشاره‌گر به ساختار FCBیی از خود فایل یا نزدیکترین دایرکتوری به فایل هدف را برمی‌گرداند.

تابع NtfsCommonCreate چک می‌کند که آیا بخشی از مسیر مورد جستجو باقی‌مانده یا نه. اگر دیگر مسیری برای جستجو باقی نمانده، تابع NtfsFindStartingNode خود فایل را پیدا می‌کند و تابع NtfsCommonCreate به پایان می‌رسد. در غیر این صورت تابع به جستجوی آن فایل در حجم مربوطه ادامه می‌دهد.

همانطور که مشاهده می‌شود تابع NtfsCommonCreate شامل حلقه‌ای است که در آن دایرکتوری‌هایی که به فایل مورد نظر هدایت می‌شوند، به ترتیب باز می‌شوند. در ابتدای حلقه بررسی می‌شود که آیا فایل جاری یک دایرکتوری هست یا خیر. اگر نبود تابع با یک خطا خاتمه می‌یابد. در غیر این صورت نام جاری از آن مسیر استخراج می‌شود و با استفاده از تابع NtfsOpenSubdirectory فایل یا دایرکتوری که آن نام را دارد باز می‌شود. تابع NtfsOpenSubdirectory یک فایل یا دایرکتوری را به صورت انحصاری نگه می‌دارد. قبل از فراخوانی NtfsOpenSubdirectory، دایرکتوری قبلی که باز بود توسط همان تابع NtfsOpenSubdirectory آزاد می‌شود. این کار حلقه تا زمان رسیدن به دایرکتوری که فایل مورد نظر در آن قرار دارد ادامه می‌یابد. در انتهای کار درصورت انجام ناموفق، تابع NtfsCommonCreate آخرین دایرکتوری پیدا شده را از طریق تابع NtfsTeardownStructures می‌بندد. همچنین این تابع شیء ERESOURCE مربوط به آن فایل یا دایرکتوری را اگر آن فایل یا دایرکتوری باز نباشد آزاد می‌کند. چون این فایل یا دایرکتوری در حال حاضر توسط سیستم فایل باز شده‌است ERESOURCE مربوط به آن آزاد شده و فایل مربوط به FCB بسته خواهد شد.

۵-۳-۱ ماهیت مشکل

هنگامی که یک تلاش برای باز کردن یک فایل با نام فایل $mft به عنوان یک دایرکتوری انجام می‌شود (مثل C:\$MFT\foo)، تابع NtfsFindStartingNode نمی‌تواند آن را پیدا کند چون این تابع کمی متفاوت جستجو می‌کند، برخلاف تابع NtfsOpenSubdirectory که فایل را همیشه پیدا می‌کند. در نتیجه چرخه کار با شروع از فایل سیستم root آغاز می‌شود. سپس تابع NtfsOpenSubdirectory فایل را باز کرده و ERESOURCE را به طور انحصاری می‌گیرد. در تکرار بعدی از حلقه مشخص می‌شود که آن فایل یک دایرکتوری نیست و بنابراین کارش را با یک خطا متوقف می‌کند و تابع NtfsCommonCreate در انتهای کارش توسط تابع NtfsTeardownStructures سعی می‌کند که آن فایل را ببندد اما با این حقیقت مواجه می‌شود که نمی‌تواند این فایل را ببندد چون این فایل توسط خود سیستم فایل باز شده است. در همان زمان برخلاف انتظار تابع NtfsCommonCreate، تابع NtfsTeardownStructures شیء ERESOURCE مربوط به $mft را تصرف کند و برای همیشه در این مرحله معلق خواهد ماند.

۵-۳-۲ قابلیت بهره برداری

بهره برداری از این آسیب‌پذیری می‌تواند از راه دور انجام گیرد. مهاجم با استفاده از این آسیب‌پذیری و دسترسی از راه دور به یک سیستم و با فریب دادن کاربر می‌تواند فایل غیر موجود با یک مسیر نادرست را باز کرده و عملیات سیستم را مختل نماید.

۵-۳-۳ سطح آسیب پذیری

CVSS ریسک آسیب‌پذیری‌های مختلف را از ۰,۱ تا ۱۰ به عنوان درجه low تا high درجه‌بندی می‌کند. این سایت به این آسیب‌پذیری امتیاز ۶,۵ را اختصاص داده و درجه آن را High ارزیابی کرده و جدول(۱) را برای توصیف آن ارائه دادند.

جدول(۱): ویژگی‌های آسیب‌پذیری.

Vector	Complexity	Authentication	Confidentiality	Integrity	Availability
Local	High	Multiple	None	None	None
Adjacent	Medium	Single	Partial	Partial	Partial
Network	Low	None	Complete	Complete	Complete

۶ مکانیزم کارکرد

در ادامه روش استفاده از آسیب‌پذیری مذکور شرح داده می‌شود.

۶-۱ خلاصه ای از حمله با استفاده از آسیب‌پذیری منع سرویس

اگر برنامه ساده‌ای مشابه شکل(۳) را کامپایل و اجرا کنیم دسترسی به حجم مورد نظر قطع[۹] خواهد شد.

```
#include <windows.h>
#include <stdio.h>
#include <tchar.h>

int _tmain( int argc, _TCHAR* argv[] ) {

    CreateFileW( L"c:\\$mft\\<любое сочетание>", FILE_READ_ATTRIBUTES, 0, NULL, OPEN_EXISTING, 0, NULL );

    return 0;

}
```

شکل(۳): مثالی برای استفاده از آسیب‌پذیری مذکور.

اگر مهاجم با استفاده از کدی مشابه شکل سعی کند هر فایل مربوط به فایل $mft را باز کند یعنی اگر از نام فایل $mft به عنوان نام دایرکتوری استفاده کند (مثل C:\$MFT\foo) ، دسترسی به کل حجم "C" قطع خواهد شد و از آنجا که این حجم، یک حجم سیستمی است، کل سیستم قطع خواهد شد. اما اگر حجم مورد استفاده یک حجم سیستمی نباشد، فقط دسترسی به همان حجم قطع خواهد شد.

همچنین این کار می‌تواند زمانی رخ دهد که کاربر سعی کند فایل را به طور مستقیم از طریق یک فرمان Run یا سایر ابزارها باز کند یا مهاجم مسیر را به طور مخفیانه در پس زمینه یک صفحه وب به عنوان منبع تصویر یک URL بارگذاری کند.

۷ اقدامات جهت کاهش شدت آسیب پذیری

متاسفانه شرکت مایکروسافت هنوز این آسیب‌پذیری را وصله نکرده است. اما آخرین نسخه‌ی ویندوز یعنی ویندوز۱۰ فاقد این آسیب‌پذیری است. همچنین می‌توان از مرورگر Chrome به جای مرورگرهای Internet Explorer و Firefox استفاده کرد زیرا مرورگر گوگل برخلاف مرورگرهای Internet Explorer و Firefox، اجازه نمی‌دهد بارگیری تصاویر با مسیرهای نادرست مانند $MFT انجام شود.

[۹] Hanging

۸ جمع بندی و نتیجه‌گیری

در این گزارش به بررسی آسیب‌پذیری منع سرویس که منجر به از کار افتادن سیستم قربانی توسط مهاجم می‌شود پرداختیم. مهاجم می‌تواند مسیری تعریف کند که در آن از نام فایل $mft به عنوان نام دایرکتوری استفاده کرده و آن مسیر را به طور مخفیانه در پس زمینه یک صفحه وب به عنوان منبع تصویر یک URL بارگذاری کند و با این کار می‌تواند منجر به crash کردن ویندوز قربانی گردد. درنتیجه با استفاده از این آسیب‌پذیری و دسترسی از راه دور به یک سیستم می‌تواند عملیات آن را مختل نماید.

۹ منابع

[۱] http://www.securityfocus.com/bid/۹۸۷۲۹/info

[۲] http://habrahabr.net/thread/۱۴۷۵۷

[۳] http://securityaffairs.co/wordpress/۵۹۵۳۵/hacking/ntfs-bug.html

[۴]https://www.bleepingcomputer.com/news/microsoft/filesystem-bug-hangs-or-crashes-windows-۷-and-windows-۸-۱/