بسمه تعالی

مرکز مدیریت امداد و هماهنگی
عملیات رخدادهای رایانه ای
سازمان فناوری اطلاعات ایران
معاونت امنیت فضای تولید و تبادل اطلاعات

راهنمای آزمون نفوذپذیری برنامه‌های کاربردی ویندوز

آموزشی

شناسه سند ..... MaherReport_13990912
نوع سند ..... گزارش فنی
شماره نگارش ۰,۱
تاریخ نگارش ۱۳۹۹/۰۹/۱۲
طبقه‌بندی سند ..... **عادی**

تهران، خیابان شهید بهشتی، نرسیده به قائم مقام فراهانی، پلاک ۲۶۷، سازمان فناوری اطلاعات ایران

(۰۲۱) ۴۲۶۵۰۰۰۰ (۰۲۱) ۴۲۶۵۰۰۰۰ cert.ir

فهرست مطالب

۱ مقدمه

هنگامی که بحث بررسی امنیتی و آزمون نفوذ مطرح می‌گردد، معمولا بلافاصله ذهن متخصصان امنیتی متوجه برنامه‌های تحت‌وب، برنامه‌های همراه و یا نهایتا ویروس‌ها و بدافزارها می‌شود. در حالی که برنامه‌هایی نیز از دیرباز وجود داشته‌اند که در هیچکدام از این دسته‌ها قرار نمی‌گیرند. این برنامه‌ها با نصب‌شدن بر روی سیستم‌عامل کاربران و در مواردی حتی بدون هیچ‌گونه نیازی به برقراری ارتباط با دیگر سیستم‌ها، می‌توانند قابلیت‌های مورد نیاز کاربر را فراهم نمایند. با گسترش هرچه بیشتر فضای وب و تمایل توسعه‌دهندگان به طراحی برنامه‌های سازگار با وب، بررسی امنیتی این برنامه‌ها به دست فراموشی سپرده شده است. تا کنون مطلب جامعی برای آزمون نفوذ برنامه‌های نصبی گردآوری نشده است. از این‌رو در این مطلب سعی شده است تا اولا مرور مختصری به سازوکار و معماری این نوع برنامه‌ها و سپس تمامی روش‌های بررسی امنیتی آن‌ها به صورت آزمون جعبه‌سیاه صورت گیرد تا بعدها از آن بتوان به عنوان مرجعی برای آزمون‌های نفوذ این برنامه‌ها استفاده نمود. ذکر این نکته لازم است که اگرچه مطالب آتی قابل اعمال در تمامی سیستم‌عامل‌هاست، به دلیل غالب بودن برنامه‌های تحت سیستم‌عامل ویندوز تمرکز این مطلب بر روی برنامه‌های ویندوزی است.

۱-۱ دسته‌بندی برنامه‌های کاربردی از لحاظ وابستگی به کارگزار

برنامه‌های کاربردی را می‌توان به دو دسته عمده برنامه‌های کاملا وابسته به کارگزار و برنامه‌های غیروابسته به کارگزار تقسیم‌بندی کرد. برنامه‌هایی را که کاملا به کارگزار وابسته هستند برنامه‌های «کارخواه نازک» نیز می‌نامند و علت این نام‌گذاری این است که این برنامه‌ها دارای ساختار بسیار ساده‌ای بوده و تقریبا تمامی کارایی آن‌ها توسط کارگزار تعریف و تعیین می‌شود. برنامه‌های تحت‌وب و بسیاری از برنامه‌های همراه در این دسته قرار می‌گیرند که تنها شامل یک رابط کاربری ساده هستند و کارگزار مربوط به این برنامه‌ها تعیین می‌کند که برنامه دارای چه قابلیت‌هایی باشد.

در سمت مقابل برنامه‌های غیروابسته به کارگزار قرار دارند. این برنامه‌ها کارکرد خود یا کاملا از وجود یک کارگزار برای بی‌نیاز بوده و می‌توانند مستقل عمل کنند و یا ارتباطشان با کارگزار به صورت مقطعی است. این برنامه‌ها با عناوینی چون برنامه‌های «کارخواه ضخیم» نام‌گذاری می‌شوند که به این مفهوم است که بخش اعظمی از کارکرد و قابلیت‌های این نوع برنامه‌ها در سمت کاربر صورت می‌گیرد. این برنامه‌ها مدت‌های مدیدی است که مورد استفاده بوده که می‌توانند هم به صورت محلی و هم به صورت کارخواه/کارگزار مورد استفاده قرار گیرند.

به علت این‌که برنامه‌های کارخواه ضخیم از قابلیت‌های بیشتری برخوردار هستند اغلب بسیار پیچیده‌تر بوده و بنابراین روش‌های بررسی امنیتی و آزمون‌های نفوذ متفاوتی نسبت به برنامه‌های کارخواه نازک دارند. در ادامه این مطلب به روش‌های آزمون و بررسی امنیتی این نوع برنامه‌ها پرداخته خواهد شد.

۱-۲ معماری‌های معمول برنامه‌های کارخواه ضخیم

در این بخش نگاهی به انواع معماری برنامه‌های کارگزار ضخیم انداخته خواهد شد که اغلب از سوی توسعه‌دهندگان مورد استفاده قرار می‌گیرد.

۱-۲-۱ معماری دو لایه

در معماری دو لایه، برنامه کارخواه ضخیم یک ارتباط کارخواه-کارگزار را مورد استفاده قرار می‌دهد. برنامه بر روی رایانه کارخواه نصب شده و برای این‌که این برنامه بتواند کارایی داشته باشد، بایستی با یک کارگزار که اغلب کارگزار پایگاه‌داده است در ارتباط باشد. به عنوان مثال برنامه‌های حسابداری اغلب برای کارکرد خود تنها به یک کارگزار پایگاه‌داده نیاز دارند تا داده‌ها را در آنجا ذخیره و هنگام نیاز فراخوانی کنند.

۱-۲-۲ معماری سه لایه

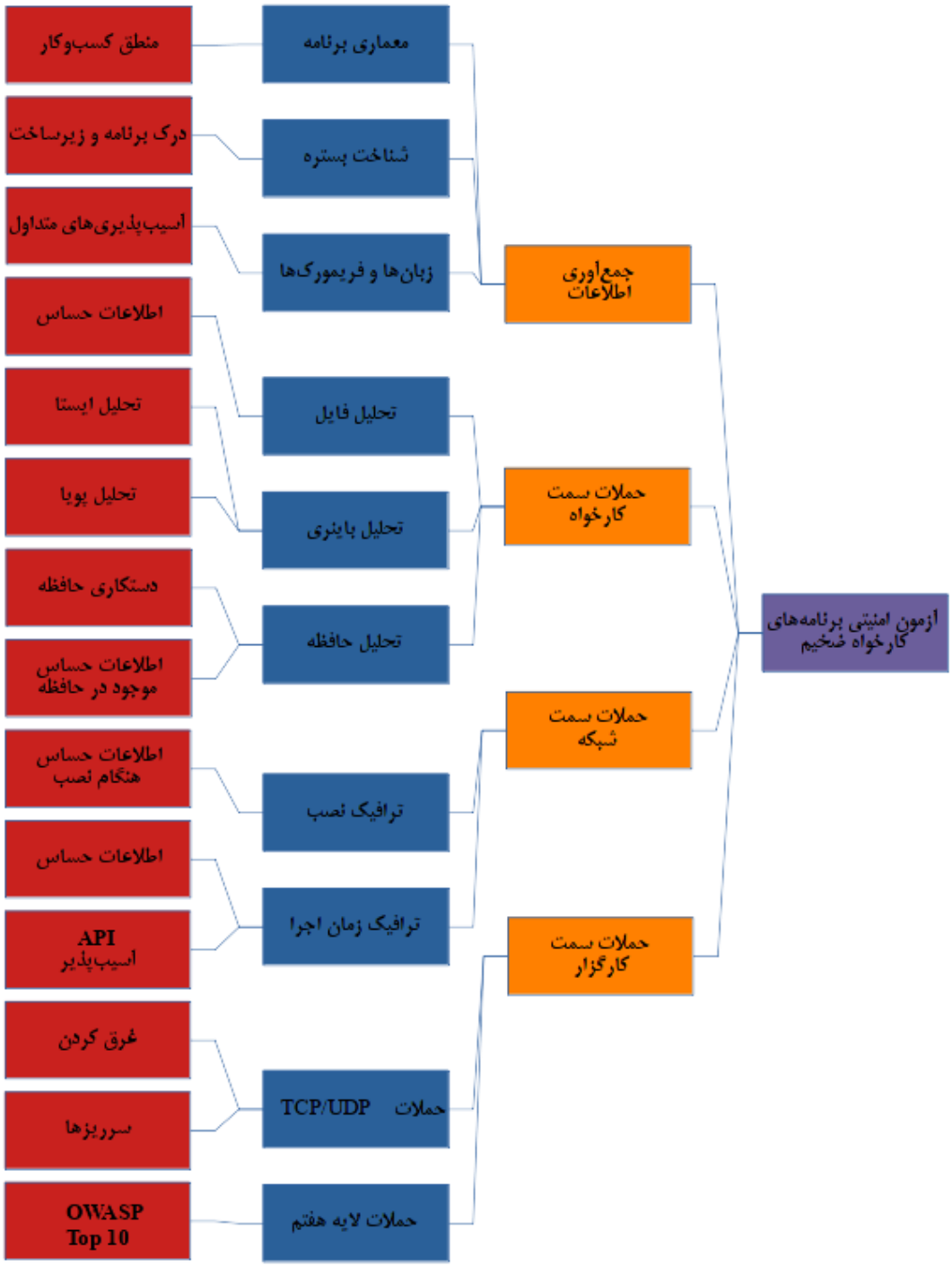
در معماری سه لایه، برنامه کارخواه با یک کارگزار برنامه در ارتباط است که مابین این برنامه کارخواه و کارگزار پایگاه‌داده قرار گرفته است. در این صورت برنامه کارخواه امکان برقراری ارتباط مستقیم با کارگزار پایگاه‌داده را نداشته و به همین علت این نوع معماری از امنیت بهتری نیز برخوردار است. معمولاً ارتباطات میان این سه لایه از طریق پروتکل‌های HTTP/HTPS صورت می‌گیرد.

۱-۳ آزمون امنیتی برنامه‌های کارخواه ضخیم

همان‌گونه که ذکر شد، برنامه‌های کارخواه ضخیم معمولاً پیچیده‌تر و همچنین متنوع‌تر از دیگر برنامه‌های تحت‌وب هستند و بنابراین نیازمند رویکردی متفاوت از نقطه‌نظر آزمون‌های نفوذپذیری هستند. فرایند آزمون‌های نفوذپذیری برنامه‌های کارخواه ضخیم را می‌توان به چهار بخش جمع‌آوری اطلاعات، حملات سمت کارخواه، حملات سمت شبکه و حملات سمت کارگزار طبقه‌بندی نمود که در بخش‌های آتی این مراحل به همراه ابزارهای قابل استفاده ذکر خواهد شد. بخش‌های اصلی تمرکز برای آزمون‌های نفوذ از پروژه «بررسی‌های امنیتی فایل‌های اجرایی باینری ویندوز» مربوط به OWASP اقتباس شده است OWASP Windows Binary Executable Files Security Checks Project اما

متأسفانه این پروژه فعلاً به اتمام نرسیده و مطالب موجود ناقص هستند. لذا در این مطلب سعی شده است تـا با گردآوری مطالب از چندین منبع مختلف تمامی ابزارها و رویه‌های مورد نیاز آزمون نفوذ برنامه‌های ویندوزی در یک مطلب جمع‌آوری شـود تا بتوان از آن به عنوان یک مرجع اسـتفاده کرد. منابع مختلفی شـامل کتاب‌ها، مقالات و صفحات وب به عنوان منابع این مطلب مورد استفاده قرار گرفته‌اند که در بخش منابع به آن‌ها اشـاره شـده اسـت. یکی از منابع بسـیار کاربردی در این زمینه کتاب «هنر بررسی امنیتی نرم‌افزار» The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities by Mark Dowd اسـت که شامل تمامی ملاحظات مورد نیاز هنگام طراحی برنامه‌های کاربردی است. همچنین در مقاله‌ای The Current and Future of Software Securities and Vulnerabilities نیز مروری بر وضـعیت کنونی و آینده بررسی امنیتی برنامه‌های کاربردی صورت گرفته است.

شـکـل زیر یـک شـمـای کلی از رونـد آزمون نفوذ برنـامـه‌های کاربردی کارخواه ضـخیم را نمایش می‌دهد OWASP Windows Binary Executable Files Security Checks Project.


آزمون امنیتی برنامه‌های کاربردی ضخیم
حملات سمت کارخواه
حملات سمت شبکه
حملات سمت کارگزار
جمع‌آوری اطلاعات
معماری برنامه
شناخت بستره
زبان‌ها و فریمورک‌ها
تحلیل فایل
تحلیل باینری
تحلیل حافظه
ترافیک نصب
ترافیک زمان اجرا
حملات TCP/UDP
حملات لایه هفتم
منطق کسب‌وکار
درک برنامه و زیرساخت
آسیب‌پذیری‌های متداول
اطلاعات حساس
تحلیل ایستا
تحلیل پویا
دستکاری حافظه
اطلاعات حساس موجود در حافظه
اطلاعات حساس هنگام نصب
اطلاعات حساس
API آسیب‌پذیر
غرق کردن
سرریزها
OWASP Top 10

۲ جمع‌آوری اطلاعات

به هنگام آزمون نفوذ امنیتی، همیشه اولین قدم جمع‌آوری اطلاعات از هدف مورد بررسی است. بنابراین برنامه‌های کارخواه ضخیم نیز از این قاعده مستثنی نیستند. در اینجا مواردی که می‌توانند در مراحل اولیه جمع‌آوری اطلاعات از هدف بسیار مفید باشند اشاره می‌شود.

۲-۱ درک کامل عملکرد و رفتار برنامه:

برای بررسی امنیتی یک برنامه بسیار مهم است که تمامی قابلیت‌ها و نحوه عمل‌کرد آن مشخص شود. هر برنامه می‌تواند دارای قابلیت‌های متفاوتی نسبت به دیگر برنامه‌ها باشد. بنابراین در وهله اول سعی می‌گردد تا نحوه عمل‌کرد تمامی اجزای برنامه کاملاً درک شود. یافتن تمامی نقاط ورودی برنامه، نوع داده‌های ورودی، وجود رابط کاربری خط فرمان، نحوه تعامل برنامه با دیگر کارگزارها، نسخه کارگزار پایگاه‌داده، نحوه مدیریت دسترسی کاربران و سطح دسترسی کاربران مختلف به بخش‌های مختلف برنامه می‌تواند کمک بسیاری در تحلیل امنیتی برنامه در مراحل بعدی بکند. به طور مثال در صورتی که یک برنامه چند کاربره است و در آن بخش مدیریت کاربران و کنترل دسترسی وجود دارد باید نحوه سطح بندی و یا مجوز دهی به کاربران درک شود و سپس بر اساس آن درستی عملکرد برنامه مورد بررسی قرار گیرد. مهمترین ابزار برای درک عملکرد برنامه استفاده از مستندات ارائه شده توسط توسعه دهنده برنامه مانند سند راهنمای کاربری برنامه و یا مستندات توصیف برنامه نام برد.

۲-۲ شناخت بستره

برنامه‌های کارخواه ضخیم عموما برای استفاده در بستره خاصی طراحی می‌شوند. به عنوان مثال برخی برنامه‌ها تنها در سیستم‌های ۳۲ بیتی و برخی دیگر هم در سیستم‌های ۳۲ بیتی و هم ۶۳ بیتی قابل استفاده هستند. دانستن این که برنامه دارای چه نیازمندی‌های سخت‌افزاری و نرم‌افزاری است می‌تواند در آزمون‌ها برای تغییر شرایط برنامه و بررسی رفتار آن در شرایط نامساعد و پیش‌بینی‌نشده توسط توسعه‌دهندگان کمک کند. بنابراین در مراحل اولیه جمع‌آوری اطلاعات باید تلاش شود تا تمامی محدودیت‌های برنامه از لحاظ نیازمندی سخت‌افزاری و نرم‌افزاری مورد تجزیه و تحلیل قرار گیرد.

۳-۲ تکنولوژی‌ها و زبان‌های مورد استفاده

برنامه‌ها تحت زبان‌های برنامه‌نویسی و چارچوب های مختلف ساخته می‌شوند. اطلاع داشتن از این که برنامه مورد بررسی توسط چه زبانی برنامه‌نویسی شده است اولاً باعث می‌گردد آسیب‌پذیری‌های معمول که مختص این نوع زبان هستند مشخص شده و مورد آزمون قرار گیرند و همچنین مهندسی معکوس قابلیت‌ها و طراحی حملات راحت‌تر صورت بگیرد.

- ابزارهای مفید

چندین ابزار برای شناسایی زبان مورد استفاده در تولید برنامه‌ها وجود دارد که به چند مورد اشاره می‌شود:

- CFF Explorer: ابزاری برای ویرایش فایل‌های اجرایی و همچنین نمایش زبان به کار رفته در تولید برنامه.

این برنامه یک ابزار بسیار قدرتمند است که دارای قابلیت‌های بسیاری است. از این برنامه می‌توان به منظور مشاهده فرایندهای درحال اجرا، تصویر گیری از حافظه سیستم، ویرایش فایل‌های اجرای و بسیاری عملیات دیگر استفاده کرد.

- PEiD : ابزاری که قادر به شناسایی انواع معمول کامپایلرها و زبان‌های برنامه‌نویسی فایل‌های اجرایی است.

فایل‌های اجرایی توسط ابزارهای خاصی تولید می‌گردند که معمولاً شامل کامپایلرها، رمزگذار ها و یا بسته‌بند ها است. هرکدام از این ابزارها امضاهای مختص خود را در فایل اجرایی از خود به جای می‌گذارند. ابزار PeiD قادر است ۴۷۰ نوع مختلف از این امضاها را کشف کند.

- Detect It Easy: ابزاری برای شناسایی نوع فایل اجرایی برای ویندوز، لینوکس و مک.

دیگر برنامه‌های مشابه برای شناسایی نوع فایل تنها به امضای فایل مراجعه می‌کنند. به علت این که این امضاها در فایل‌ها در بایت‌های مشخصی قرار دارد بنابراین این برنامه‌ها تنها آن بایت‌ها را بررسی می‌کنند و امکان اضافه نمودن دیگر پارامترها وجود ندارد. در نتیجه امکان تشخیص اشتباه نوع فایل وجود دارد. برنامه DEI می‌تواند با دخیل کردن دیگر پارامترها و همچنین داشتن قابلیت کدنویسی و اضافه کردن افزونه توسط کاربر مقدار خطا در تشخیص فایل‌ها را کم می‌کند.

۲-۴ فایل‌های سیستمی درگیر هنگام اجرای برنامه

هنگام اجرای برنامه‌ها معمولاً برخی فایل‌های سیستمی سیستم‌عامل نیز مورد استفاده قرار می‌گیرند. دانستن اینکه چه فایل‌هایی از سیستم عامل مورد استفاده برنامه قرار می‌گیرند می‌تواند در بعضی موارد باعث افشای اطلاعات مهم و حساسی باشد. در بخش‌های بعدی شرح داده خواهد شد که کتابخانه‌های لینک پویا یا همان فایل‌های DLL که برای استفاده برنامه از سیستم‌عامل فراخوانی می‌شوند می‌توانند مورد سواستفاده برای برخی حملات قرار گیرند. لذا جمع‌آوری اطلاعات در این زمینه بسیار مفید خواهد بود. برای این منظور می‌توان از ابزار زیر کمک گرفت.

- ابزارهای مفید

Process Monitor: یک ابزار دیگر از بسته ابزاری SysInternals که برنامه‌ها را به همراه کتابخانه‌ها و فایل‌های سیستمی درحال استفاده توسط آن برنامه نمایش می‌دهد.

این ابزار به صورت ترکیبی از ۲ ابزار معروف Filemon و Regmon است که قابلیت‌های بیشتری نیز به آن افزوده شده است. توسط این برنامه می‌توان به جزئیات بسیاری در خصوص فرایندهای درحال اجرا و فایل‌ها و تنظیمات رجیستری در حال استفاده و خواندن‌ها نوشتن‌های جاری را به صورت زنده ملاحظه کرد.

۲-۵ فایل‌های خاص مورد نیاز برنامه برای اجرا

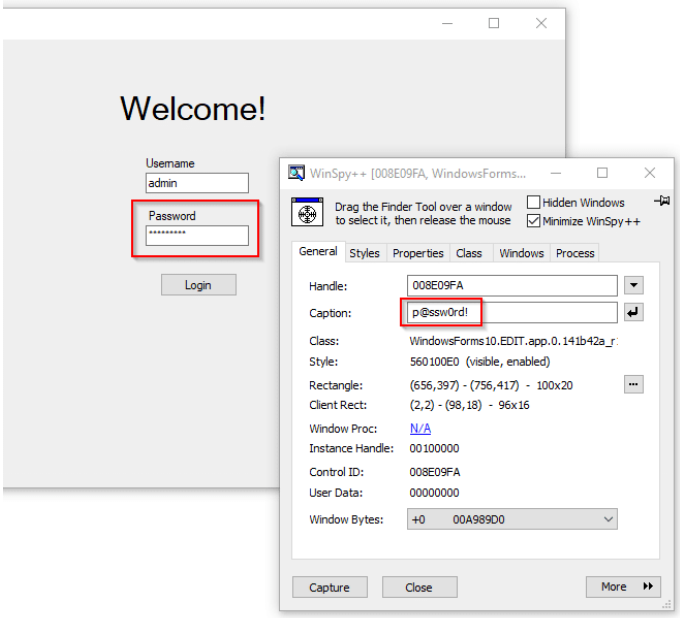
بسیاری از برنامه‌ها دارای یک فایل تنظیمات اصلی هستند که برای اجرا نیازمند مراجعه به اطلاعات ثبت‌شده داخل این فایل هستند. این فایل‌ها بعضی اوقات می‌توانند حاوی اطلاعات حساسی باشند و یا بعضا می‌توان با اعمال تغییراتی در این فایل‌ها رفتار برنامه را تغییر داده و اعمال مخربی بر روی برنامه اجرا کرد و یا مراحل امنیتی برنامه را دور زد. برای یافتن این فایل‌ها بایستی پوشه محل نصب برنامه، نقاطی که برنامه در آنجا فایل‌های موقتی تولید و ذخیره می‌کند و درنهایت هر نقطه‌ای از سیستم که برنامه در آنجا ردپایی داشته باشد مورد بررسی قرار گیرد. به طور مثال بسیاری از فایل‌های اجرایی شامل یک فایل به نام Config.ini هستند. این فایل را می‌توان به وسیله ویرایشگر متن باز کرده و گاهی اوقات به اطلاعات مفیدی دست یافت. برای یافتن این نوع فایل‌ها به هر فایلی که به نظر می‌رسد شامل تنظیمات برنامه است می‌توان مشکوک شده و مورد بازبینی قرار داد. همچنین به وسیله ابزارهایی مانند Process Monitor و Filemon می‌توان برنامه را اجرا کرده و فایل‌های غیرسیستمی را که در هنگام اجرای برنامه بارگذاری و خوانده می‌شوند یافته و ارزیابی کرد.

۳ حملات سمت کارخواه

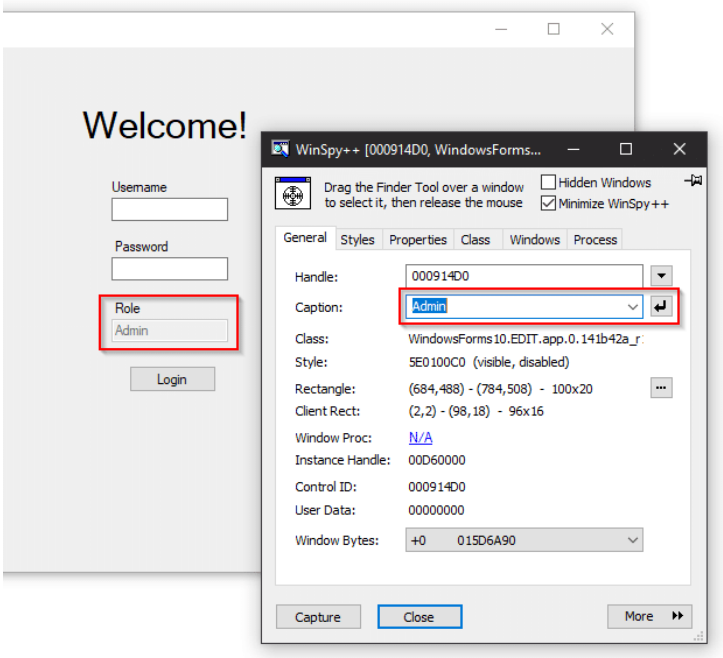
تمامی برنامه‌های کاربردی یک رابط کاربری برای برقراری ارتباط کاربر با قابلیت‌های برنامه دارند. از سوی دیگر اشاره شد که در برنامه‌های کارخواه ضخیم بسیاری از قابلیت‌های برنامه در سمت کاربر قرار داده‌شده است و بنابراین این نوع برنامه‌ها دارای سطح حمله وسیع‌تر بوده و مهاجمان می‌توانند حملات گسترده‌تری را علیه برنامه پیاده کنند. از این‌رو در بخش‌های بعدی به بررسی حملات ممکن از سمت کارخواه و معرفی ابزارهای مفید پرداخته خواهد شد.

۳-۱ رابط کاربری گرافیکی

بیشتر برنامه‌های کارخواه ضخیم از رابط کاربری گرافیکی استفاده می‌کنند. همان‌گونه که قبلاً گفته شد تمامی نقاط ورودی برنامه بایستی مورد ارزیابی قرار گیرد. برنامه های مختلف از روش‌ها و کتابخانه های متفاوتی برای ایجاد رابط کاربر استفاده می کنند. نکته مهمی که در همه این موارد وجود دارد این است که عموما نمی توان برای کنترل دسترسی به رابط کاربری اعتماد کرد. برای مثال برنامهها غالباً دارای یک فرم ورود کاربران با دسترسی‌های مختلف هستند. برخی اوقات این فرم‌ها قابلیت ذخیره رمز عبور را در داخل فرم دارند. با ابزارهایی همچون WinSpy می‌توان مقادیر داخل این کادرها را که در حالت عادی مخفی هستند به نمایش درآورده و استخراج کرد. در شکل زیر نمونه‌ای از استفاده از این برنامه برای نمایش رمزعبور ذخیره‌شده در برنامه نمایش داده شده است.



بعضی برنامه‌ها شامل یک بخش غیر قابل تغییر برای تعیین نقش کاربران هستند که در حالت عادی و با کلیک کردن نمی‌توان آن را تغییر داد. با استفاده از ابزار WinSpy همان‌طور که در شکل زیر نیز مشاهده می‌شود می‌توان این کار را نیز انجام داد.



در بسیاری مواقع پس از ورود توسط کاربر سطح پایین؛ برخی از قابلیت‌های برنامه مانند گزینه‌های منوها یا دکمه‌ها و برگه‌ها برای کاربر غیرفعال می‌شود (زیرا که مخصوص کاربران با سطح دسترسی بالاتر هستند). با استفاده از این برنامه یا برنامه‌های مشابه می‌توان موارد غیرفعال‌شده در رابط گرافیکی را به راحتی فعال کرد.

- ابزارهای مفید

به کمک ابزارهای زیر می‌توان رابط گرافیکی برنامه‌ها را برای تغییر این قابلیت‌ها مورد آزمون قرار داد:

- WinSpy: یک برنامه برای مشاهده جزئیات پنجره‌های باز در ویندوز

همان‌گونه که مثال‌هایی ارائه شد، به کمک این برنامه می‌توان نام پنجره‌ها، اطلاعات محتوای داخل آن‌ها، کلاس پنجره‌ها، رمزهای عبور و بسیاری از جزئیات دیگر از پنجره‌های فعال را مشاهده کرد.

- Window Detective: ابزاری برای مشاهده و همچنین ویرایش جزئیات پنجره‌های درحال اجرا

این ابزار بسیار مشابه ابزار قبلی است که به علاوه تمامی ساختار زیرمجموعه‌های پنجره‌ها را به شکل درختی نمایش می‌دهد و یافتن عناصر و المان‌ها را در داخل پنجره‌ها آسان می‌کند.

- WinManipulate: ابزاری ساده و متن‌باز برای دستکاری اشیای موجود در پنجره‌های برنامه‌ها

به کمک این برنامه می‌توان اشیای پنجره‌های ویندوز همچون دکمه‌ها، منوها، جعبه‌های متن و کنترل‌کننده‌های خاص را دستکاری کرده و عمل‌کرد آن‌ها را تغییر داد.

- Windows Enabler: یک برنامه ساده که قابلیت‌های غیرفعال برنامه‌ها را فعال می‌سازد.

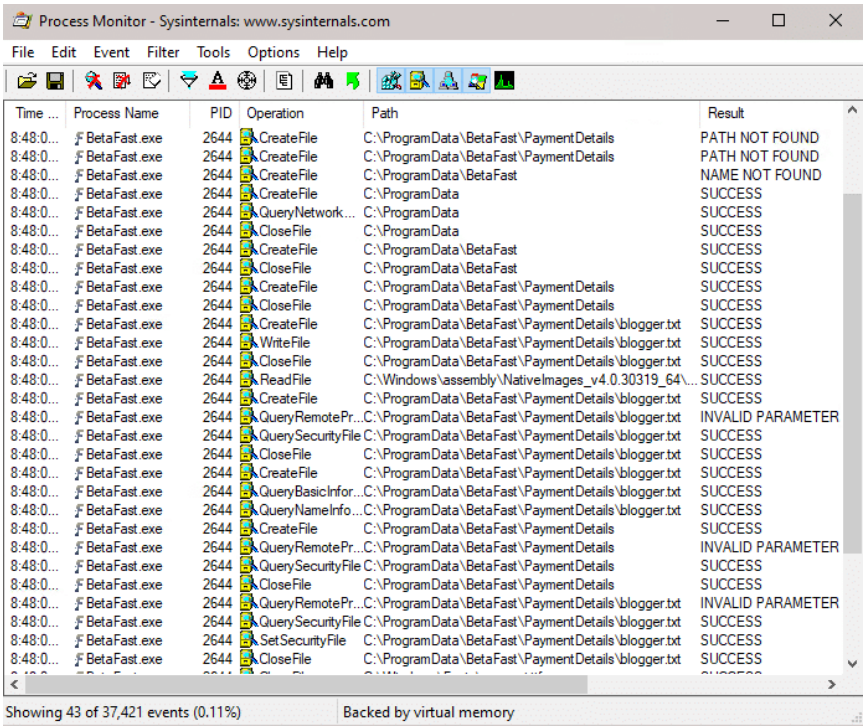
این برنامه پس از نصب و اجرا یک آیکون کوچک در قسمت tray سیستم نمایش می‌دهد. استفاده از این برنامه بسیار آسان است. برای فعال‌سازی منوی خاصی از برنامه، کافی است بر روی آیکون یاد شده کلیک شود تا کلمه «ON» بر روی آیکون نمایش داده شود. به این ترتیب منوی مورد نظر فعال شده و قابل استفاده خواهد بود.

- Snoop: ابزار متن‌باز برای جاسوسی/جستجو در ساختار درختی گرافیکی برنامه‌ها، تغییر جزئیات، و بسیاری قابلیت‌های دیگر

این برنامه برای جاسوسی در برنامه‌های WPF طراحی شده است. برنامه‌های ساخته‌شده به صورت WPF پیچیده‌تر از برنامه‌های ساده هستند اما در عین حال امکان طراحی‌های بسیار متنوع‌تری را ارائه می‌کنند. برای دستکاری قابلیت‌های این برنامه‌ها، ابزارهای قبلی اشاره شده ناتوان هستند و باید از چنین ابزاری که مختص این نوع برنامه‌هاست استفاده گردد. برای مطالعه بیشتر در این خصوص می‌توان به مرجع Introduction to Hacking Thick Clients: Part 1 – the GUI مراجعه نمود.

۳-۲ تحلیل فایل

برنامه‌ها معمولاً اطلاعاتی را در فایل‌های محلی و رجیستری سیستم ذخیره می‌کنند. در بخش قبلی اشاره شد که با ابزارهایی همچون Process Monitor می‌توان فایل‌هایی که هنگام اجرای برنامه فراخوانی می‌شوند را استخراج کرد. با تحلیل این فایل‌ها می‌توان اطلاعات حساسی همچون رمزهای عبور و آدرس‌های دوردست اتصال برنامه به کارگزار را پیدا کرد. به طور نمونه در شکل زیر توسط ابزار Process Monitor مشاهده می‌شود که برنامه به طور مرتب با فایل blogger.txt در ارتباط است. بنابراین این فایل احتمالاً حاوی اطلاعات خاصی است. پس از باز کردن محتویات این فایل مشاهده می‌کنیم که این فایل حاوی اطلاعات کارت بانکی کاربر به صورت رمزنگاری‌نشده است.



در بعضی مواقع توسعه دهندگان این اطلاعات را به صورت رمزنگاری‌شده ذخیره می‌کنند که در این حالت باید اولاً به ضعف در فرایند رمزنگاری شک کرده و آن را بررسی کرد و همچنین احتمال یافتن کلید رمزنگاری در یک نقطه دیگر از فایل‌های داخل سیستم وجود دارد بنابراین مثلاً پایگاه‌داده یا دیگر نقاط مشکوک برای یافتن چنین کلیدهایی بررسی شود.

یکی از مهم‌ترین موارد در تحلیل فایل، فایل نصبی برنامه است. اگر فایل نصبی برنامه در دسترس باشد، باید تمامی نقاطی که برنامه هنگام نصب عملیات خواندن و نوشتن انجام می‌دهد مورد دقت قرار گیرد. برخی اطلاعات هنگام نصب در جایی ذخیره می‌شوند و بعداً توسط برنامه خوانده می‌شوند. مثلاً ممکن است اطلاعات اتصال به پایگاه‌داده در رجیستری سیستم ذخیره شود یا نام کاربری و رمز هبور پیش‌فرض برنامه جایی در میان فایل‌های سیستمی نوشته شود.

- ابزارهای مفید

از ابزارهای زیر برای تحقیق و تحلیل در خصوص عملیات در حال اجرا توسط برنامه می‌توان استفاده کرد:

- Process Monitor: ابزاری قدرتمند برای مشاهده عملیات درحال اجرا توسط برنامه‌ها با قابلیت فیلتر کردن توسط دسته‌بندی‌های مختلف
- AccessEnum: ابزاری برای مشاهده کامل فایل سیستم و رجیستری

نحوه کار این برنامه به این صورت است که این برنامه با استفاده از API امنیتی ویندوز به اطلاعات خواندن، نوشتن و رد دسترسی‌ها در برنامه‌ها دسترسی یافته و در داخل برنامه آن‌ها را نمایش می‌دهد.

- Regshot: یک ابزار متن‌باز برای گرفتن عکس‌فوری از رجیستری ویندوز

برای این که بتوان تشخیص داد یک برنامه هنگام اجرا چه تغییراتی را در سیستم رجیستری ویندوز اعمال می‌کند، بهترین روش می‌تواند مقایسه حالت‌های قبل و بعد از استفاده از یک برنامه در رجیستری باشد. بنابراین این ابزار می‌تواند با ذخیره تمامی اطلاعات رجیستری در این ۲ حالت و مقایسه آن‌ها و در یافتن تغییرات اعمال‌شده توسط برنامه بخصوص کمک کند.

- Process Explorer: نوع پیشرفته و کامل‌تر برنامه مدیریت برنامه‌های ویندوز

قابلیت‌های منحصر به فرد این برنامه آن را قادر به ردیابی مشکلات و درزهای اطلاعاتی مربوط به نسخه‌های DLL کرده و همچنین درک کاملی از نحوه عمل‌کرد ویندوز و برنامه‌ها را ممکن می‌کند.

- Process Hacker: ابزار قدرتمند چند منظوره دیگری برای مشاهده عملیات درحال اجرا، اشکال‌زدایی برنامه‌ها و کشف بدافزارها

این ابزار دارای قابلیت‌های بسیار فراوان است که از آن‌ها می‌توان به مشاهده جزئیات فعالیت‌های برنامه‌ها، دسترسی برنامه‌ها به دیسک، مشاهده زنده استفاده از شبکه، ردیابی استفاده شدن فایل بخصوص توسط فرایند خاص و بسیاری از قابلیت‌های دیگر اشاره کرد که مخصوصاً در تحلیل و ردیابی فایل‌ها کاملاً سودمند است.

۳-۳ آسیب‌پذیری دزدیدن فایل‌های DLL

برنامه‌هایی که تحت سیستم‌عامل ویندوز اجرا می‌شوند از کتابخانه‌های از قبل آماده DLL برای بسیاری از کارکردهای خود کمک می‌گیرند. دزدیدن DLL یک نوع حمله است که با قرار دادن فایل‌های DLL دلخواه در مسیری که برنامه هنگام اجرا برای بارگزاری فایل DLL مورد نظر خود جستجو می‌کند، فایل DLL مورد نظر مهاجم به جای فایل اصلی مورد استفاده قرار می‌گیرد و به این ترتیب کدهای دلخواه مهاجم از داخل این فایل اجرا می‌شوند.

- ابزارهای مفید

برای بررسی این آسیب‌پذیری می‌توان از ابزارهای زیر کمک گرفت:

- DLLSpy: ابزاری برای کشف دزدیدن DLL در فرایندهای در حال اجرا و فایل‌های باینری

این ابزار یک برنامه متن‌باز است که در سیستم‌های ویندوز ۷ به بالا کاربرد دارد. نحوه عمل‌کرد آن به این صورت است که این برنامه دارای ۳ موتور ایستا، پایا و بازگشتی است که می‌تواند توسط کاربر فعال یا غیرفعال شود. این موتورها با جستجو در داخل فرایندهای در حال اجرا در سیستم و در داخل فایل‌های DLL که در سیستم بارگزاری شده‌اند، و سپس با بررسی این که در نقاطی که این فایل‌ها وجود دارند امکان نوشتن و رونویسی فایل وجود دارد یا خیر، امکان دزدیدن DLL را کشف و گزارش می‌کند.

- Robber: یک ابزار متن‌باز برای یافتن فایل‌های اجرایی مستعد حملات دزدیدن DLL

این ابزار نیز همانند برنامه بالا امکان دزدیدن DLL را پیدا می‌کند و از الگوریتم ساده‌تری برای این منظور استفاده می‌کند.

۳-۴ تحلیل باینری

تحلیل فایل‌های باینری یک مبحث بسیار گسترده است که تعداد بیشماری از آسیب‌پذیری‌هایی که روزانه گزارش می‌شوند از این طریق به‌دست می‌آیند. تحلیل باینری می‌تواند به ۲ صورت تحلیل ایستا و تحلیل پویا انجام گیرد که در حالت تحلیل ایستا سعی می‌گردد تا به کمک تجزیه‌گرها دستورات و توابع به‌کار رفته در برنامه مهندسی معکوس شده و مورد تحلیل و بررسی قرار گیرند. در حالی که در تحلیل پویا برنامه اجرا شده و به کمک اشکال‌زدا ها مقادیر متغیرها در حافظه مورد تحقیق و بررسی و دستکاری قرار می‌گیرد تا آسیب‌پذیری‌هایی همچون حملات سرریز بافر کشف گردد. با استفاده از تحلیل باینری امکان کشف اطلاعاتی همچون موارد زیر وجود دارد:

- ☑ اطلاعات حساس مانند نام‌کاربری و رمزعبور قرار گرفته داخل کد برنامه هنگام برنامه‌نویسی
- ☑ کلیدها و رمزهای مورد استفاده در API
- ☑ آدرس‌های مورد استفاده در API
- ☑ علکردهای مخفی
- ابزارهای مفید

همان‌گونه که گفته شد بحث تحلیل باینری بسیار مفصل بوده و خارج از محدوده این مطلب است بنابراین در این‌جا تنها به ذکر چند ابزار مفید در این زمینه بسنده می‌شود.

- ☐ Interactive Disassembler(IDA Pro): یکی از کامل‌ترین برنامه‌های تجزیه‌گر قابل اجرا بر روی بسیاری از بسترها

به کمک این برنامه که در اصل یک تحلیل‌گر ایستای فایل‌های باینری است می‌توان دستورالعمل‌های به کار رفته در فایل باینری را به زبانی قابل خواندن توسط انسان که به آن زبان اسمبلی گفته می‌شود تبدیل کرد تا مورد تحلیل و بررسی قرار گیرند. این برنامه همچنین دارای ابزار اشکال‌زدا است که قابلیت‌های تحلیل ایستای این برنامه را تکمیل کرده و به کاربر اجازه می‌دهد در داخل دستورالعمل‌ها قدم بگذارد و تحقیقات عمیق‌تری روی فایل باینری داشته باشد.

- ☐ Ghidra: ابزار متن‌باز قدرتمند جهت مهندسی معکوس برنامه‌ها

این برنامه توسط تیم تحقیقاتی سازمان NSA توسعه یافته و پشتیبانی می‌شود که قابلیت‌های متعددی همچون تجزیه‌کردن کد، دیکامپایل، گراف بندی و اسکریپت‌نویسی در زبان‌های جاوا و پایتون را دارد. این برنامه جدید که روز به روز به تعداد طرفداران آن نیز افزوده می‌شود یک رقیب جدی برای برنامه IDA است.

- Immunity Debugger: ابزاری برای تحلیل پویای برنامه‌ها با قابلیت‌های بسیار و افزونه‌های بیشمار

این ابزار به طور تخصصی توسط محققان امنیتی برای طراحی و طرح‌ریزی اکسپلویت ها مورد استفاده قرار می‌گیرد. این برنامه که هم شامل رابط گرافیکی و هم رابط خط فرمان است امکان تحلیل فایل‌های باینری را فراهم کرده و دستورالعمل‌ها را به شکل زبان اسمبلی نشان می‌دهد و علاوه بر آن به صورت یک اشکال‌زدای قدرتمند امکان تحلیل مقدار متغیرها در هر قدم از اجرای برنامه را می‌دهد. این برنامه نیز با دارا بودن هسته اسکریپت‌نویسی بسیار قوی امکان اجرای هرنوع دستور دلخواه را فراهم کرده و همچنین موجب توسعه افزونه‌ها و ابزارهای بسیار برای این ابزار شده است.

- Ollydbg: تجزیه‌گر فایل‌های باینری ۳۲ بیتی ویندوزی

این برنامه تحلیل کدهای باینری بسیار سبک و بدون نیاز به نصب و در عین حال دارای قابلیت‌هایی از جمله قابلیت تحلیل انواع مختلف فایل‌های باینری، ردیابی ثبات ها، شناخت حلقه‌ها و فراخوانی‌های API، اشکال‌زدایی برنامه‌های چندرشته‌ای و قابلیت‌های متعدد دیگر است.

- Radare2: فریمورک متن‌باز مهندسی معکوس برای بسیاری از پلتفرم‌ها

این ابزار متن‌باز نیز یکی دیگر از ابزارهایی است که می‌توان از آن برای مهندسی معکوس فایل‌های باینری، اشکال‌زدایی توسط هسته‌های متعدد و وصله‌زنی برنامه‌ها برای افزودن قابلیت‌ها یا اصلاح آسیب‌پذیری‌ها استفاده کرد.

- dnSpy: ابزاری برای مهندسی معکوس فایل‌های اجرایی ساخته‌شده توسط زبان .NET

فایل‌های اجرایی که توسط زبان‌برنامه‌نویسی .NET ساخته شده‌اند، قابلیت دیکامپایل شدن دارند و بنابراین با استفاده از چنین ابزاری خواندن کدها، اشکال‌زدایی و تحلیل رفتار این نوع فایل‌ها بسیار راحت‌تر از فایل‌های نوشته‌شده با زبان‌های دیگر همچون C خواهند بود.

- x64dbg: ابزاری برای تجزیه فایل‌های باینری ویندوزی ۳۲ و ۶۴ بیتی

این برنامه نیز سعی کرده است تا بسیاری از قابلیت‌های ابزارهای معروف تحلیل باینری را در خود گنجانده و در عین حال با حجم کم و همچنین متن‌باز بودن برای همگان قابل استفاده باشد.

- JetBrains DotPeek: ابزاری رایگان برای تجزیه و تحلیل فایل‌های ساخته‌شده با زبان .NET

یک برنامه دیگر که توسط شرکت معروف JetBrains به منظور دیکامپایل برنامه‌های تولیدشده با زبان برنامه‌نویسی .NET ایجاد شده است و دارای قابلیت‌های فراوان همچون گرفتن خروجی از فایل دیکامپایل شده به انواع دیگر فایل قابل انطباق با پروژه‌های ویژوال استودیو است.

- ILSpy: برنامه متن‌باز تجزیه‌گر فایل‌های .NET.

به کمک این برنامه می‌توان علاوه بر دیکامپایل کردن فایل‌های اجرایی .NET. تمامی پروژه را نیز می‌توان دیکامپایل کرد. همچنین این برنامه روش‌های پیمایش و جستجوی مختلفی را برای سهولت تجزیه و تحلیل کدها در اختیار قرار می‌دهد.

- JD-GUI: ابزاری برای مهندسی معکوس فایل‌های جاوا

فایل‌های جاوایی که به شکل CLASS توسط کامپایلرهای جاوا تولید می‌شوند نیز قابل خواندن به صورت عادی نیستند. اما با استفاده از این ابزار می‌توان سورس کدهای این فایل‌ها را استخراج کرده و مورد مطالعه قرار داد.

- JADX: ابزاری برای استخراج کدهای جاوا از فایل‌های Dex و APK

این برنامه نیز یک ابزار بسیار قدرتمند است که قابلیت استخراج تقریباً هرگونه فایل اجرایی تولید شده توسط زبان‌های مرتبط با جاوا را دارد.

- PE Explorer: ابزاری برای مشاهده، ویرایش و مهندسی معکوس فایل‌های اجرایی

این ابزار قادر به ویرایش انواع مختلف فایل‌های اجرای از جمله فایل‌های با پسوند EXE و DLL و بسیاری پسوندهای دیگر را دارد و همچنین پشتیبانی مخصوص از فایل‌های نوشته شده توسط زبان دلفی را شامل می‌گردد.

- Frida: یک جعبه‌ابزار قدرتمند جهت مهندسی معکوس سازگار با تمامی بسترها

قدرت اصلی این ابزار در قابلیت اعمال تغییرات دلخواه در برنامه‌های مختلف است. به طور مثال برای افزودن یک قابلیت به یک برنامه، نیازی به از نو نوشتن آن برنامه نیست و می‌توان توسط این ابزار و در زبان اسکریپت‌نویسی دلخواه همچون پایتون یا جاوااسکریپت این قابلیت را به برنامه اضافه کرد. بنابراین از چنین ابزاری می‌توان در حملات نیز استفاده کرده و کارایی‌های دلخواهی را بر روی برنامه اعمال کرد.

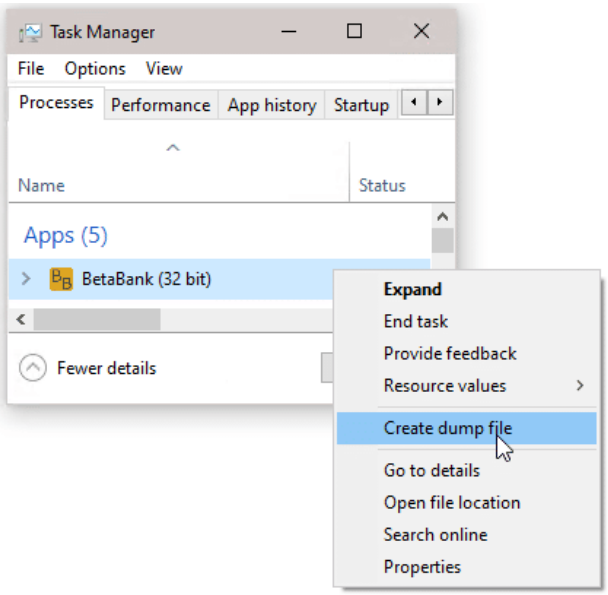
- Strings: ابزاری برای استخراج تمامی رشته‌ها به صورت UNICOE یا ASCII در داخل فایل‌های اجرایی

این برنامه یکی از ساده‌ترین و در عین حال پرکاربردترین برنامه‌ها هنگام بررسی فایل‌های باینری است. فایل‌های اجرایی و باینری به علت این که به صورت رمزگذاری‌شده هستند قابل خواندن توسط ویرایشگرهای متن نیستند. اما توسط این ابزار می‌توان تمامی متن‌هایی که در داخل فایل اجرایی به صورت رشته باقی مانده‌اند استخراج کند. این متن‌ها می‌توانند رمزهای عبور داخل کد، آدرس‌های اتصال و غیره باشند که توسط این برنامه استخراج شده و در حملات مورد استفاده قرار می‌گیرند.

۳-۵ تحلیل حافظه

هنگام بررسی آسیب‌پذیری‌های یک برنامه، بایستی یک تحلیل حافظه از زمانی که برنامه در حال اجراست صورت گیرد. با تحلیل حافظه می‌توان به ۲ هدف رسید. اول دست‌یابی به اطلاعات حساس که توسط برنامه در حافظه موقت ذخیره شده‌اند و دوم دستکاری حافظه برای دست‌یابی به قابلیت‌های دلخواه از برنامه.

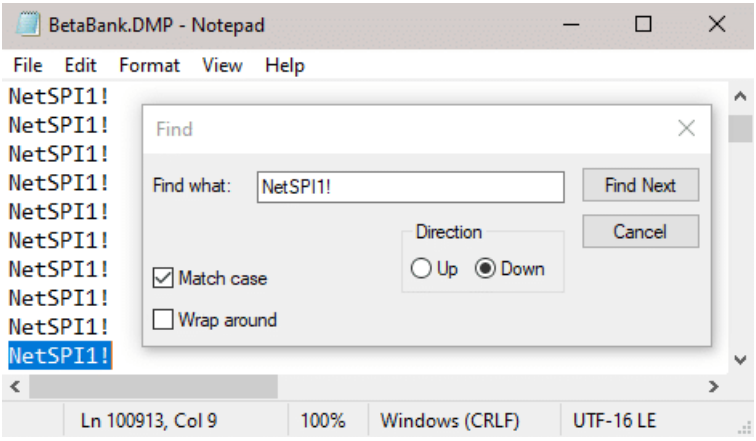
برنامه‌هایی که از معماری دو ردیفی استفاده می‌کنند یک عیب طراحی وجود دارد و آن این است که این برنامه‌ها اطلاعات حساس را از طریق حافظه برنامه منتقل می‌کنند و بنابراین با بررسی حافظه هنگامی که برنامه در حال اجراست، می‌توان به این اطلاعات دست یافت. در بعضی مواقع نیز برنامه هنگام اجرا متغیرهایی را در حافظه ذخیره می‌کند که این مقادیر پس از بسته‌شدن برنامه نیز همچنان در حافظه وجود داشته و قابل استخراج هستند. برای بررسی اطلاعات داخل حافظه می‌توان از ابزارهایی که کل حافظه سیستم در حال حاضر را در یک فایل ذخیره می‌کنند استفاده کرده و سپس در داخل آن فایل به جستجوی اطلاعات مورد نظر پرداخت. ساده‌ترین برنامه برای ذخیره حافظه سیستم در فایل، مدیریت برنامه‌های ویندوز است و لذا بدون نیاز به نصب ابزار خاصی می‌توان این کار را انجام داد. در شکل زیر مشاهده می‌شود که از داخل برنامه مدیریت برنامه‌های ویندوز با کلیک راست بر روی فرایند در حال اجرا، می‌توان حافظه موقت سیستم را در یک فایل ذخیره کرد.



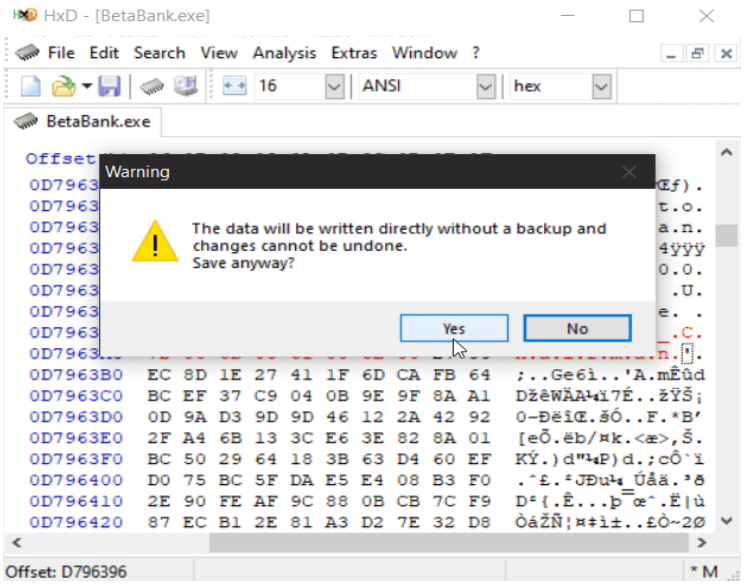
پس از این که تمامی حافظه در یک فایل ذخیره شد می‌توان اطلاعات داخل آن را خواند. به علت این که فایل ایجاد شده دارای حجم بالایی است و خواندن آن مشکل است و همچنین این فایل دارای مقادیر غیر رشته‌ای بسیاری است، یک راه می‌تواند استفاده از ابزار String به منظور استخراج داده‌های رشته‌ای با طول دلخواه باشد. به عنوان مثال با دستوری مانند دستور زیر تمامی رشته‌های داخل فایل تولید شده در مرحله قبلی با نام BetaBank.DMP استخراج شده و در فایلی دیگر ذخیره می‌شوند:

strings.exe .\BetaBank.DMP | Out-File -FilePath .\BetaBank.DMP.txt\.

سپس با باز کردن این فایل در یک ویرایشگر متن ساده، در شکل زیر ملاحظه می‌شود که رمز عبور کاربر در داخل فایل پیدا می‌شود.



همچنین همان‌گونه که ذکر شد، تحلیل حافظه می‌تواند به منظور دستکاری آن نیز به کار گرفته شود. به عنوان مثال هنگامی که برنامه در حال اجراست، دستورات و توابعی در حافظه قرار می‌گیرند تا هنگام اجرا اعمالی را اجرا کنند. مثلاً یک دستور SQL که به پایگاه‌داده ارسال می‌شود تا داده‌ای را که مربوط به کاربری که هم‌اکنون وارد برنامه شده است تغییر دهد. در این حالت‌ها می‌توان با برنامه‌هایی همچون HxD که قابلیت ویرایش حافظه به صورت زنده را نیز دارد استفاده کرده و حافظه موقت سیستم را دستکاری کرد تا از برنامه سو استفاده شود. شکل زیر مربوط به مثالی است که در آن یک برنامه رشته‌های دستور SQL حاوی اطلاعات کاربر وارد شده به برنامه را به پایگاه‌داده ارسال می‌کند. مهم‌ترین قسمت این دستورات بخشی است که مربوط به دریافت مبلغی تعیین‌شده توسط کاربر از داخل برنامه است. در این حالت با استفاده از ابزار HxD می‌توان این دستورات را یافته و نام کاربر را با کاربر دلخواه جایگزین کرده و در نتیجه مبلغ مورد نظر از حساب شخصی دیگر کسر گردد.



- ابزارهای مفید

به طور کلی برنامه‌ها می‌توانند در سطح وسیعی برای آسیب‌پذیری‌های سمت حافظه در خطر باشند و بنابراین بررسی حافظه یک قدم بسیار مهم در تحلیل برنامه‌هاست. ابزارهای کاربردی که به این منظور می‌توانند سودمند باشند اشاره می‌شود:

- HxD: یک ابزار ویرایش حافظه، دیسک و فایل‌ها

این برنامه در اصل یک ویرایشگر hex یا همان مبنای ۱۶ متن و فایل است که علاوه بر آن قابلیت ویرایش دیسک و حافظه نیز در آن گنجانده شده است. از قابلیت‌های خاص این برنامه می‌توان به توانایی آن برای باز کردن و ویرایش فایل‌های بسیار حجیم با سرعت بسیار بالا اشاره کرد.

- Cheat Engine: یکی از معروف‌ترین برنامه‌های ویرایش و دستکاری بازی‌ها و برنامه‌ها

این برنامه از دیرباز مورد استفاده کاربران بازی‌های رایانه‌ای برای اعمال تغییرات در ساختار بازی‌ها قرار می‌گرفته است. این برنامه دارای یک پویش‌گر حافظه سیستم است تا متغیرهای به کار رفته در داخل برنامه یا بازی را شناسایی کرده و به کاربر امکان تغییر آن‌ها را فراهم کند. همچنین این برنامه مجهز به بسیاری از ابزارهای مهندسی معکوس همچون اشکال‌زداها و تجزیه‌گرهاست که آن را به ابزاری کامل برای استفاده در تحلیل‌های نرم‌افزاری تبدیل می‌کند.

- Winhex: ابزاری پیشرفته برای ملاحظه و ویرایش فعالیت‌ها و رجیستری و حافظه به صورت زنده

این ابزار نیز همانند ابزار قبلی علاوه بر ویرایشگر hex دارای قابلیت‌های ویرایش دیسک و حافظه و همچنین مقایسه و تحلیل داده هاست.

- Volatility: یک ابزار متن‌باز مقایسه رجیستری

درواقع تمرکز اصلی این برنامه بر روی استخراج داده‌ها و اطلاعات از حافظه موقت سیستم است اما دارای چندین ابزار مختلف نیز بوده و همچنین به علت ساخته شدن توسط زبان پایتون قابل استفاده در بسیاری از بسترهاست.

۴ حملات سمت شبکه

قبلاً توضیح داده شد که برنامه‌های کارخواه ضخیم می‌توانند به دو صورت برنامه‌های دو لایه و برنامه‌های سه لایه طراحی شوند. از لحاظ بررسی برنامه تحت شبکه تفاوت این دو نوع طراحی برنامه نوع پروتکل ارتباطی است. به عنوان نمونه یک برنامه طراحی‌شده به صورت دو لایه به علت این که تنها با کارگزار پایگاه‌داده در ارتباط است لذا تنها از پروتکل‌های تعیین‌شده به این منظور استفاده خواهد کرد. در حالی که یک برنامه طراحی‌شده به صورت سه لایه می‌تواند از چندین پروتکل شامل پروتکل‌های وب نیز استفاده کند. به هرحال می‌توان با استفاده از ابزارهای مفید در این زمینه تمامی ارتباطات برنامه دیده‌بانی شده و مورد تجزیه و تحلیل قرار گیرد. با بررسی این ارتباطات ممکن است اطلاعات زیر کسب شود:

- ☑ اطلاعات حساس منتقل‌شده در کانال ارتباطی غیر رمزنگاری‌شده(شامل رمزهای عبور، کلیدهای API و غیره)
- ☑ نقاط انتهایی وب همچون وب‌سرویس‌ها
- ☑ فایل‌های ارسالی تحت شبکه
- ☑ پروتکل‌های مورد استفاده توسط برنامه

۴-۱ ترافیک هنگام نصب

قبلاً نیز ذکر شد که در صورت داشتن دسترسی به فایل نصبی برنامه می‌توان اطلاعات مفیدی از آن کسب کرد. همان‌گونه که هنگام اجرای فایل نصبی ممکن است این برنامه اطلاعات حساسی را در رجیستری و یا فایل‌های سیستمی ذخیره‌کند، ممکن است این برنامه با اتصال به شبکه نیز تعاملاتی با کارگزار برنامه و یا کارگزار پایگاه‌داده داشته باشد. بنابراین می‌توان با استفاده از ابزارهای نظارت شبکه همچون Wireshark و اجرای فایل نصبی، ارتباطات احتمالی برنامه با دیگر نقاط را تحلیل کرد.

۲-۴ ترافیک هنگام اجرا

برنامه‌های کارخواه ضخیم اگر نیازمند مراجعه به کارگزار باشند به هر حال زمانی از اجرای برنامه به آن مراجعه خواهند کرد. در برنامه‌های دو ردیفی این ارتباط به صورت دستورات مربوط به خواندن و نوشتن از پایگاه‌داده است. با اجرای برنامه‌های دیده‌بانی همانند Wireshark می‌توان این دستورات را دریافت کرده و خواند. اگر داده‌های حساسی از این تعاملات به صورت رمزنگاری‌نشده باشند می‌توان با بازکردن بسته‌های انتقالی آن‌ها را خواند. یک ابزار بسیار مفید دیگر در این زمینه می‌تواند Echo Mirage باشد که نه تنها قابلیت شنود اطلاعات بلکه ویرایش آن‌ها به صورت زنده را نیز داراست. با استفاده از چنین ابزاری در بعضی مواقع می‌توان پس از انتخاب یک گزینه از منوی برنامه و سپس ویرایش بعضی مقادیر در میانه راه، عملیات مخربی را بر روی برنامه انجام داد.

در صورتی که برنامه از نوع سه لایه باشد، برای بررسی ترافیک شبکه‌ای آن می‌توان از ابزار قدرتمند Burp Suite استفاده کرد که برای حملات وب بسیار معروف است. این ابزار به صورت یک پراکسی در میان کارخواه و کارگزار تنظیم می‌گردد تا تمامی ارتباطات برنامه با کارگزار از داخل آن عبور کرده و قابل ویرایش باشند. برنامه‌های سه لایه از دیدگاه قابلیت تنظیم پراکسی می‌توانند به ۲ نوع «دارای قابلیت تنظیم پراکسی» و «بدون قابلیت تنظیم پراکسی» تقسیم‌بندی گردند. به این معنی که در برنامه‌های دارای قابلیت تنظیم پراکسی، در داخل خود برنامه یک بخش مخصوص به این منظور اختصاص داده شده است که به راحتی می‌توان در آن قسمت برنامه Burp Suite را به عنوان یک سرور پراکسی تعریف کرده و بررسی‌ها را ادامه داد. در صورتی که برنامه از چنین قابلیتی برخوردار نباشد، می‌توان ترافیک کل سیستم یا به عبارتی ترافیک رابط شبکه در حال استفاده توسط برنامه را از داخل این پراکسی عبور داد. همچنین این برنامه دارای یک قابلیت به نام «پراکسی کردن نامرئی » نیز به همین منظور است.

مورد دیگری که هنگام بررسی ترافیک یک برنامه هنگام اجرا باید به آن توجه گردد، نقاط API است که احتمالاً برنامه با آن‌ها در ارتباط است. نقاط API به علت این که توسعه‌دهندگان تصور می‌کنند که از دید کاربران مخفی خواهد ماند، می‌توانند از نظر امنیتی از دید توسعه دهندگان پنهان مانده و لذا حاوی آسیب‌پذیری‌های فراوان باشند. یک ابزار بسیار مفید برای این منظور برنامه API Monitor است که قابلیت مشاهده و کنترل فراخوانی‌های API را دارد و از آن می‌توان برای مشاهده تمامی جزئیات فراخوانی‌های API استفاده کرده و آزمون‌های نفوذ را بر روی آن‌ها امتحان کرد.

- ابزارهای مفید

ابزارهای زیر در زمینه تجزیه و تحلیل و طراحی حملات در ببستر شبکه مورد استفاده قرار می‌گیرند:

- TCPView: ابزار مشاهده ترافیک موجود در شبکه

این برنامه از دسته ابزارهای SysInternals است که تمامی نقاط انتهایی موجود در سیستم که از پروتکل‌های TCP و UDP چه به صورت محلی و چه به صورت دوردست استفاده می‌کنند نمایش می‌دهد.

- Tcpdump: برنامه خط فرمان مشاهده ترافیک شبکه

یک ابزار ساده که در محیط خط‌فرمان استفاده می‌شود، تمامی تعاملات در داخل شبکه را با جزئیات نمایش می‌دهد.

- Wireshark: ابزاری برای گرفتن بسته‌های انتقالی در داخل شبکه و باز کردن و تحلیل آن‌ها

این ابزار را می‌توان قدرتمندترین و پر استفاده‌ترین ابزار در زمینه مشاهده جزئیات فعالیت‌های موجود بر روی شبکه نامید. این برنامه بسیار مشابه برنامه Tcpdump بوده ولی دارای رابط کاربری گرافیکی بوده که در آن فیلترهای متعدد برای نمایش بسته‌ها، قابلیت خروجی گرفتن از محتوای برنامه و حتی داده‌های خام درحال انتقال توسط پورت‌های یواس‌بی نیز قرار داده شده است.

- Echo Mirage: ابزاری برای مداخله در ترافیک TCP و ویرایش آن.

برای استفاده از این ابزار می‌توان برنامه‌ای را پس از اجرا به آن متصل کرد. در این صورت ترافیک انتقالی از برنامه مورد ارزیابی از این ابزار عبور کرده و امکان ویرایش بسته‌های انتقالی را می‌دهد. از قابلیت‌های این ابزار می‌توان به رویدادنگاری آن و همچنین استفاده از قواعد دلخواه برای گرفتن بسته‌ها نام برد.

- Smartsniff: ابزاری دیگر به منظور مشاهده و گرفتن و ویرایش بسته‌های انتقالی در داخل شبکه

این ابزار از ۳ روش مختلف برای گرفتن بسته‌های انتقالی در سیستم‌های ویندوزی استفاده می‌کند. یکی از این روش‌ها استفاده از درایوری به نام WinPcap است که در محیط‌های ویندوزی برای دسترسی به لایه‌های زیرین شبکه نیاز بوده و برخی از ابزارهای دیگر از آن استفاده نکرده و بعضا در دریافت بسته‌ها با مشکل مواجه می‌شوند.

- Microsoft Network Monitor: ابزاری دیگر برای تحلیل ترافیک شبکه و پروتکل‌ها

این ابزار یکی از ابزارهای قدیمی شرکت مایکروسافت به منظور مشاهده و مدیریت ترافیک شبکه است که نسخه ۳ آن قابلیت‌های بیشتری نسبت به نسخه‌های گذشته دارد.

- API Monitor: برنامه‌ای به منظور مشاهده و تحلیل فراخوانی‌های API برنامه‌ها و کتابخانه‌ها

هنگامی که یک برنامه از قابلیت‌های API برای رسیدن به اهداف خود استفاده می‌کند، بهترین ابزار برای استفاده برنامه API Monitor است. این ابزار تمامی فراخوانی‌های API صورت گرفته توسط برنامه را به نمایش گذاشته و با دارا بودن پایگاه‌داده عظیمی از انواع تعاریف و دستورات مورد استفاده در APIها، قادر به نمایش محتویات فراخوانی‌ها با جزئیات بسیار است که تحلیل آن‌ها را بسیار راحت‌تر می‌سازد.

۵ حملات سمت کارگزار

پس از این که تمامی مراحل قبلی پشت‌سر گذاشته شد، می‌توان به حملات سمت کارگزار پرداخت. این حملات از نظر روش عمل هیچ‌گونه تفاوتی با حملات معمول روزمره که بر روی شبکه و یا در وب صورت می‌گیرند ندارند. تمامی حملاتی که بر روی شبکه و بر روی کارگزارهای دوردست می‌توان انجام داد را می‌توان در این قسمت امتحان کرد. به عنوان مثال حملات انکار سرویس بر روی پروتکل‌های TCP یا UDP و یا انواع سرریزهای بافر رامی‌توان بر روی کارگزار آزماش نمود. یکی از برجسته‌ترین نوع حملات در این بخش، حملات تزریق کد در سمت سرور مانند حمله تزریق کد SQL است که به دلیل استفاده تقریباً تمامی برنامه‌های کارخواه ضخیم از کارگزار پایگاه‌داده می‌تواند مورد تمرکز بیشتری قرار گیرد. اما در حالت کلی و خصوصا اگر برنامه از نوع سه ردیفی بوده و از یک کارگزار مختص برنامه استفاده کند، تمامی حملات مختص وب در اینجا کاربرد خواهند داشت. به علت گستردگی بخش حملات لایه هفتم، در اینجا به آن‌ها پرداخته نخواهد شد و به مرجع مختص خود در OWASP Top 10 Web Application Vulnerabilities ارجاع داده می‌شود.

نکته‌ی مهمی که در اینجا باید اشاره شود این است که در برنامه های کاربردی تحت وب به علت یکسان بودن روش دریافت و ارسال ورودی کاربر امکان بررسی بسیاری از آسیب پذیری ها مانند آسیب‌پذیری های تزریق به کمک ابزارهایی مانند Nesus ممکن است. ولی در برنامه های کارخواه ضخیم به علت تنوع در رابط های گرافیکی و نحوه دریافت و ارسال ورودی‌های کاربر، بررسی این حملات بسیار سخت تر از برنامه های کابردی تحت وب است.

۶ منابع

[1] OWASP, OWASP Windows Binary Executable Files Security Checks Project (https://wiki.owasp.org/index.php/OWASP_Windows_Binary_Executable_Files_Security_Checks_Project)

[2] Mark Dowd, The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities by Mark Dowd, 2006

[3] Cuixue Zhang, Meijiao Zhou, Yalian Xie, Xiangli Li, The Current and Future of Software Securities and Vulnerabilities ,2014

[4] Austin Altmann, Introduction to Hacking Thick Clients: Part 1 – the GUI (https://blog.netspi.com/introduction-to-hacking-thick-clients-part-1-the-gui/)

[5]OWASP, OWASP Top 10 Web Application Vulnerabilities (https://owasp.org/www-project-top-ten/2017)