

Buer Loader Found in an Unusual Email Attachment

labs.vipre.com/buer-loader-found-in-an-unusual-email-attachment/

Posted by VIPRE Labs

The COVID-19 pandemic has resulted in people ramping up online activities working from home, online shopping and relying more on online services. Recently, we came across a spam email lurking in the wild. This spam email is disguised as a known logistics company and has an unusual attachment. Malicious attackers trick the victim into believing that the email is legitimate by using a legitimate domain in the sender's email address. The content is also properly constructed and also uses a known logo making it difficult to spot that it is a malicious email.

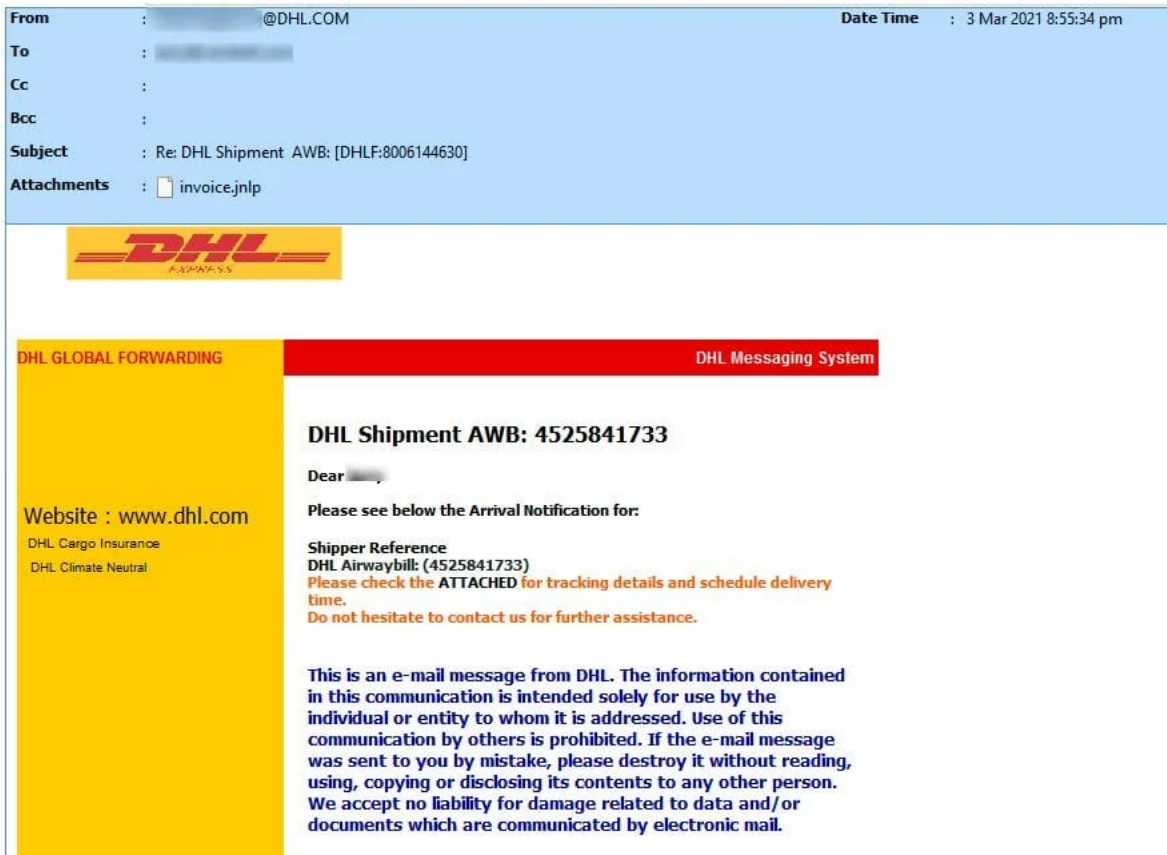


Figure 1.0 Spam email with .jnlp attachment

As threats become more prominent, we should always be cautious. These are some indicators that will show that this email is suspicious and not legitimate:

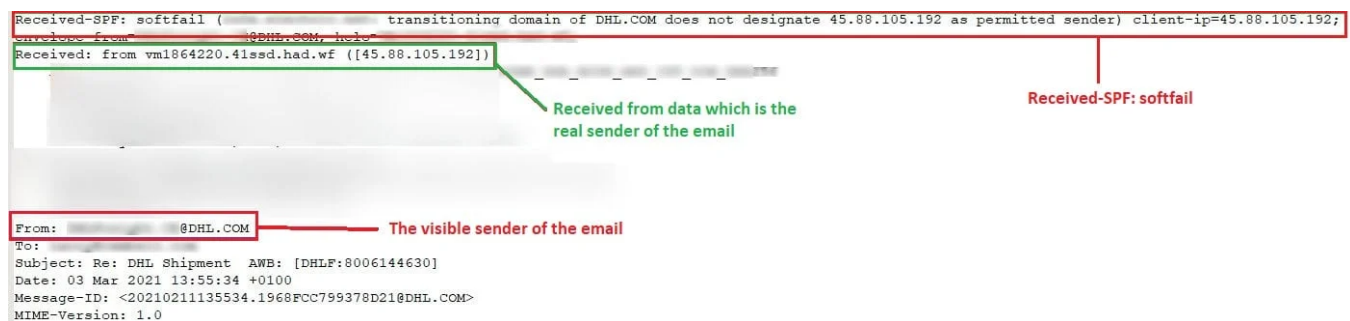


Figure 2.0 The email header

- Checking the email header, we can see that the “received from” which is in the green box in Figure 2.0, didn’t match with the “from” field (the visible sender of the email). The “received from” data is the most reliable and it is where we can see the real sender of the email. Upon researching, the domain in the “received from” header is not related to DHL. With this, the email header is forged.

- An Additional checker is the Received-SPF: softfail. It says that the “domain of DHL.COM does not designate 45.88.105.192 as permitted sender”. Upon checking, the IP address 45.88.105.192 in the “received from” is not also related to DHL.
- The attachment of the email is a .jnlp file is a Java Network Launch Protocol which is an unusual attachment for an email.

Analyzing the attachment

We will now proceed on the analysis of the jnlp file attachment that has a filename “invoice.jnlp”. We said earlier that .jnlp stands for Java Network Launch Protocol, that’s used for launching java applications from a hosted web server on a remote desktop client. Checking the jnlp file, we can see that the file will download invoice.jar from a web server hxxp://invoicsecure[.]net/documents when executed.

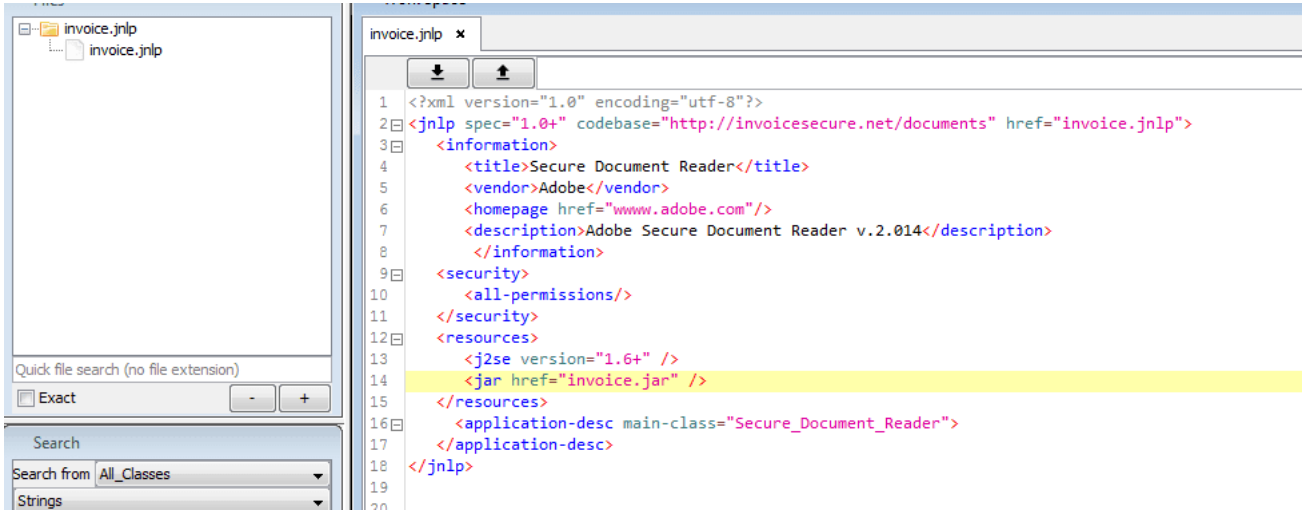


Figure 3.0 The jnlp file

The downloaded file is an invoice.jar file which is a Java Archive file. When we tried to launch the file it will show this output:

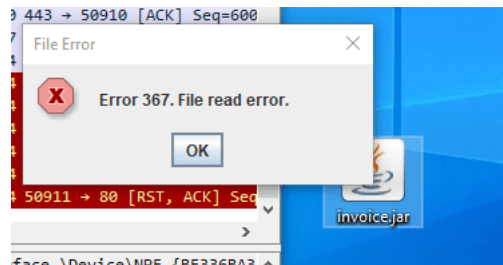


Figure 4.0 The error message upon launching invoice.jar

With this message, the victim will think that it was an error and will ignore the file. But upon analyzing the invoice.jar, we found out that this message is just a decoy. The attackers just made this technique to trick their victims and make the malware run without suspicion. Based on its code after showing an error message, it will start to read the data from “hxxp://invoicsecure[.]net/img/footer[.]jpg” and saved it as “C:\ProgramData\drvvr32.exe”. Then use Desktop.getDesktop().open() to open drvvr32.exe.

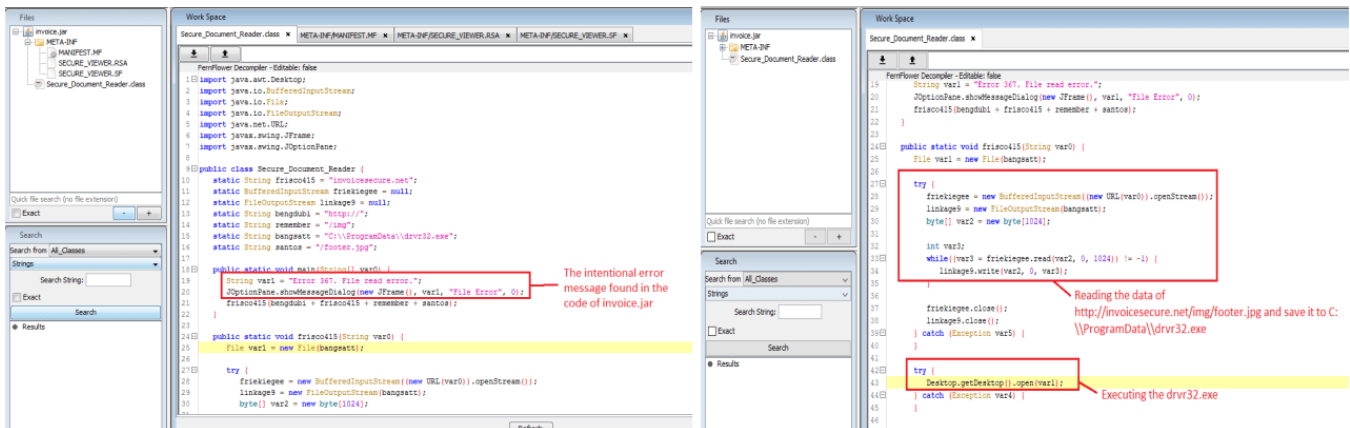
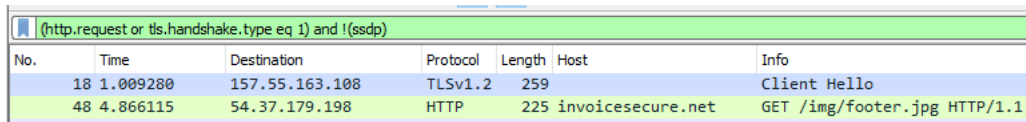


Figure 5.0 The decompiled invoice.jar



No.	Time	Destination	Protocol	Length	Host	Info
18	1.009280	157.55.163.108	TLSv1.2	259		Client Hello
48	4.866115	54.37.179.198	HTTP	225	invoicesecure.net	GET /img/footer.jpg HTTP/1.1

Figure 6.0 The HTTP GET Request once invoice.jar was executed

The Buer Loader

The malicious downloaded file was named “drv32.exe” and disguised as a legitimate xls viewer application:

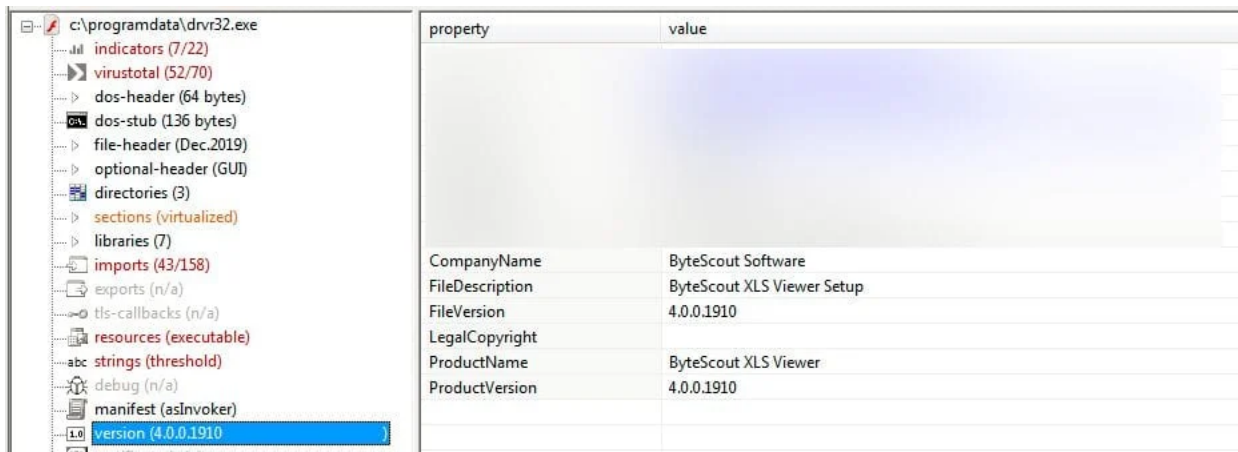


Figure 7.0 Disguising as a legitimate file

This file was identified as a type of a malware loader known as Buer Loader. This loader was first seen in 2019 and commonly distributed through malicious spam email campaigns.

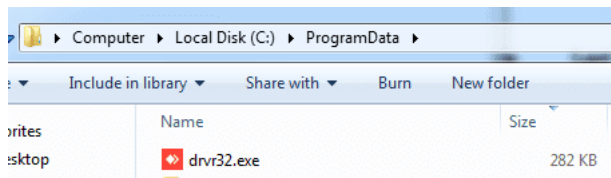


Figure 8.0 The buer loader

When executed, it will first create its installation folder “zsadsadsad” at the Startup folder and create a copy of itself in %AppData%. The created folder “zsadsadsad” contains LNK shortcut file. We decoded the LNK file to analyze all the available information it contains and we found out that it will link to the created copy.

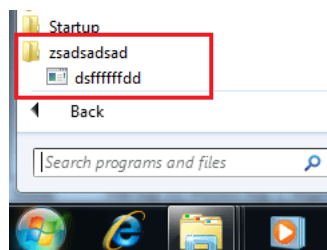


Figure 9.0 The installation folder “zsadsadsad” and the Ink shortcut file

```

File size: 0
Flags: HasTargetIdList, HasRelativePath, HasIconLocation, IsUnicode, HasExpIcon
File attributes: 0
Icon index: 0
Show window: SwNormal (Activates and displays the window. The window is restored to its original size and position if the window is minimized or maximized.)
Relative Path: ..\..\..\..\xcvcxzcxcz.exe
Icon Location: C:\Users\tst\AppData\Roaming\xcvcxzcxcz.exe
--- Target ID information (Format: Type ==> Value) ---
Absolute path: My Computer\C:\Users\tst\AppData\Roaming\xcvcxzcxcz.exe the created copy
-Root folder: GUID ==> My Computer
-Drive letter ==> C:

```

Figure 10.0 The decoded information of LNK file linking to the created copy

Throughout our analysis, we found out that this loader has an anti-analysis. It will check if the following DLLs are existing in the place where the malware is running:

E8 5CD2FFFF	call 4000000buermmodified.3F8B2F83	
8BC	mov ecx,esi	
8945 D0	mov dword ptr ss:[ebp-30],eax	[ebp-30]:L"avghooka.dll"
E8 22D2FFFF	call 4000000buermmodified.3F8B2F83	
8BC	mov ecx,esi	
8945 D4	mov dword ptr ss:[ebp-2C],eax	[ebp-2C]:L"avghookx.dll"
E8 60D2FFFF	call 4000000buermmodified.3F8B2FC8	
8BC	mov ecx,esi	
8945 D8	mov dword ptr ss:[ebp-28],eax	[ebp-28]:L"snxhk.dll"
E8 6ED2FFFF	call 4000000buermmodified.3F8B2FE3	
8BC	mov ecx,esi	
8945 DC	mov dword ptr ss:[ebp-24],eax	[ebp-24]:L"sbied11.dll"
E8 7FD2FFFF	call 4000000buermmodified.3F8B2FFE	
8BC	mov ecx,esi	
8945 E0	mov dword ptr ss:[ebp-20],eax	[ebp-20]:L"dbghe1p.dll"
E8 8DD2FFFF	call 4000000buermmodified.3F8B3016	
8BC	mov ecx,esi	
8945 E4	mov dword ptr ss:[ebp-1C],eax	[ebp-1C]:L"api_log.dll"
E8 98D2FFFF	call 4000000buermmodified.3F8B302E	
8BC	mov ecx,esi	
8945 E8	mov dword ptr ss:[ebp-18],eax	[ebp-18]:L"dir_watch.dll"
E8 A9D2FFFF	call 4000000buermmodified.3F8B3046	
8BC	mov ecx,esi	
8945 EC	mov dword ptr ss:[ebp-14],eax	[ebp-14]:L"pstorec.dll"
E8 87D2FFFF	call 4000000buermmodified.3F8B305E	
8BC	mov ecx,esi	
8945 F0	mov dword ptr ss:[ebp-10],eax	[ebp-10]:L"vmcheck.dll"
E8 C5D2FFFF	call 4000000buermmodified.3F8B3076	
8BC	mov ecx,esi	
8945 F4	mov dword ptr ss:[ebp-C],eax	[ebp-C]:L"wpespy.dll"
E8 D3D2FFFF	call 4000000buermmodified.3F8B308E	
8BC	mov ecx,esi	
8945 F8	mov dword ptr ss:[ebp-8],eax	[ebp-8]:L"cmdvrt64.dll"
E8 E1D2FFFF	call 4000000buermmodified.3F8B30A6	
8945 FC	mov dword ptr ss:[ebp-4],eax	[ebp-4]:L"cmdvrt32.dll"
775	xor esi,esi	

Figure 11.0 The DLLs to check

As per checking, some of the checked DLLs above are related to anti-virus and debuggers.

Then, it will call functions like GetCurrentHwProfileA, GetComputerNameW, and GetVolumeInformation to collect the information of the infected machine. The collected information will be combined in an allocated memory and will be formatted using sprintfw function.

33F6	xor esi,esi	
0FB64435 C4	movzx eax,byte ptr ss:[ebp+esi-3C]	
50	push eax	eax:L"cb9f1daa"
68 68628B3F	push 4000000buermmodified.3F8B6268	3F8B6268:L"%02x"
57	push edi	
FF15 1C60BB3	call dword ptr ds:[<&sprintfw>]	
83C4 0C	add esp,C	
57	push edi	
FF73 04	push dword ptr ds:[ebx+4]	[ebx+4]:L"cb9f1daa"
85F6	test esi,esi	
75 07	jne 4000000buermmodified.3F8B4238	
E8 C6E8FFFF	call 4000000buermmodified.3F8B2AFC	
E8 05	jmp 4000000buermmodified.3F8B423D	
E8 38E9FFFF	call 4000000buermmodified.3F8B2B75	
46	inc esi	
83FE 20	cmp esi,20	20: ' '
7C D1	j1 4000000buermmodified.3F8B4214	
5E	pop esi	

Figure 12.0 Routine for formatting the collected information

The collected display name, globally unique identifier (GUID) string for the hardware profile, and the computer name

0012BFFC	01970000	L" {WIN-SPF5F65H243-29539-1798} {e29ac6c0-7037-61de-816d-306e6f6e6963} Undocked_Profile"
0012BFF0	0012D76C	
0012BFF4	AA1D9FC8	
0012BFF8	C04505CC	
0012BFFC	670CBFE5	
0012C000	D214D6EA	

The formatted information

Address	Hex	ASCII
01A70000	63 00 62 00 39 00 66 00 31 00 64 00 61 00 61 00	c.b.9.f.1.d.a.a.
01A70010	63 00 63 00 30 00 35 00 34 00 35 00 63 00 30 00	c.c.0.5.4.5.c.0.
01A70020	65 00 35 00 62 00 66 00 30 00 63 00 36 00 37 00	e.5.b.f.0.c.6.7.
01A70030	65 00 61 00 64 00 36 00 31 00 34 00 64 00 32 00	e.a.d.6.1.4.d.2.
01A70040	34 00 38 00 38 00 34 00 35 00 37 00 30 00 38 00	4.8.8.4.5.7.0.8.
01A70050	38 00 39 00 63 00 66 00 37 00 34 00 32 00 35 00	8.9.c.f.7.4.3.5.
01A70060	35 00 37 00 32 00 66 00 61 00 35 00 35 00 65 00	5.7.2.f.a.5.5.e.
01A70070	31 00 33 00 34 00 62 00 31 00 30 00 31 00 32 00	1.3.4.b.1.0.1.2.

Figure 13.0 The formatted string of victim's machine information

After this, it will call other functions to retrieve more information of the infected machine and to use these information for the malware's next actions:

- RtlGetVersion
- GetNativeSystemInfo
- GetComputerNameW
- GetDriveTypeA
- GetDiskFreeSpaceExA
- GetUserNameW
- NetWkstaGetInfo

All of the other retrieved information will be combined to the formatted string above and the output is this:

cb9f1daacc0545c0e5bf0c67ead614d24884570889cf7435572fa55e134b1012|bc31re1bs5a8d1fc4ddb3cc4b75594c31b8c00de3fdfa31fgg1ad7|x32|1|User|WIN-SPF5F5SH244|14/59|WORKGROUP|test|0

Then, this output will be formatted again using `wsprintfw` function and the result is this:

02940000	33 00 33 00 61 00 36 00 63 00 35 00 31 00 39 00	3.3.a.6.c.5.1.9.
02940010	64 00 64 00 64 00 65 00 34 00 31 00 30 00 36 00	d.d.d.e.4.1.0.6.
02940020	66 00 63 00 36 00 35 00 66 00 65 00 33 00 34 00	f.c.6.5.f.e.3.4.
02940030	66 00 34 00 65 00 65 00 39 00 36 00 62 00 63 00	f.4.e.e.9.6.b.c.
02940040	33 00 38 00 35 00 62 00 31 00 63 00 63 00 32 00	3.8.5.b.1.c.c.2.
02940050	31 00 61 00 66 00 36 00 36 00 63 00 30 00 62 00	1.a.f.6.6.c.0.b.
02940060	61 00 62 00 65 00 61 00 61 00 36 00 61 00 33 00	a.b.e.a.a.6.a.3.
02940070	37 00 36 00 62 00 66 00 36 00 65 00 31 00 35 00	7.6.b.f.6.e.1.5.
02940080	35 00 38 00 65 00 62 00 61 00 33 00 33 00 66 00	5.8.e.b.a.3.3.f.
02940090	64 00 64 00 62 00 38 00 61 00 33 00 38 00 32 00	d.d.b.8.a.3.8.2.
029400A0	63 00 30 00 65 00 37 00 35 00 64 00 63 00 63 00	c.0.e.7.5.d.c.c.
029400B0	31 00 35 00 38 00 61 00 64 00 61 00 32 00 66 00	1.5.8.a.d.a.2.f.
029400C0	62 00 65 00 64 00 61 00 62 00 34 00 61 00 34 00	b.e.d.a.b.4.a.4.
029400D0	30 00 32 00 35 00 33 00 65 00 39 00 35 00 62 00	0.2.5.3.e.9.5.b.
029400E0	34 00 36 00 61 00 33 00 34 00 38 00 38 00 31 00	4.6.a.3.4.8.8.1.
029400F0	62 00 66 00 63 00 35 00 37 00 30 00 33 00 34 00	b.f.c.5.7.0.3.4.
02940100	64 00 61 00 33 00 37 00 38 00 63 00 62 00 39 00	d.a.3.7.8.c.b.9.
02940110	35 00 33 00 30 00 62 00 36 00 38 00 34 00 34 00	5.3.0.b.6.8.4.4.
02940120	36 00 36 00 63 00 30 00 32 00 30 00 31 00 38 00	6.6.c.0.2.0.1.8.
02940130	62 00 36 00 64 00 61 00 34 00 37 00 65 00 33 00	b.6.d.a.4.7.e.3.
02940140	33 00 33 00 35 00 32 00 62 00 39 00 63 00 64 00	3.3.5.2.b.9.c.d.
02940150	36 00 39 00 35 00 37 00 32 00 66 00 61 00 37 00	6.9.5.7.2.f.a.7.
02940160	33 00 38 00 34 00 61 00 61 00 33 00 63 00 37 00	3.8.4.a.a.3.c.7.
02940170	65 00 37 00 30 00 31 00 63 00 30 00 34 00 61 00	e.7.0.1.c.0.4.a.
02940180	34 00 32 00 32 00 62 00 33 00 35 00 31 00 66 00	4.2.2.b.3.5.1.f.

Figure 14.0 The 2nd round formatting of string of the victim's machine information

After retrieving and formatting the needed information of the victim's machine, Buer Loader will make it to a base64 string:

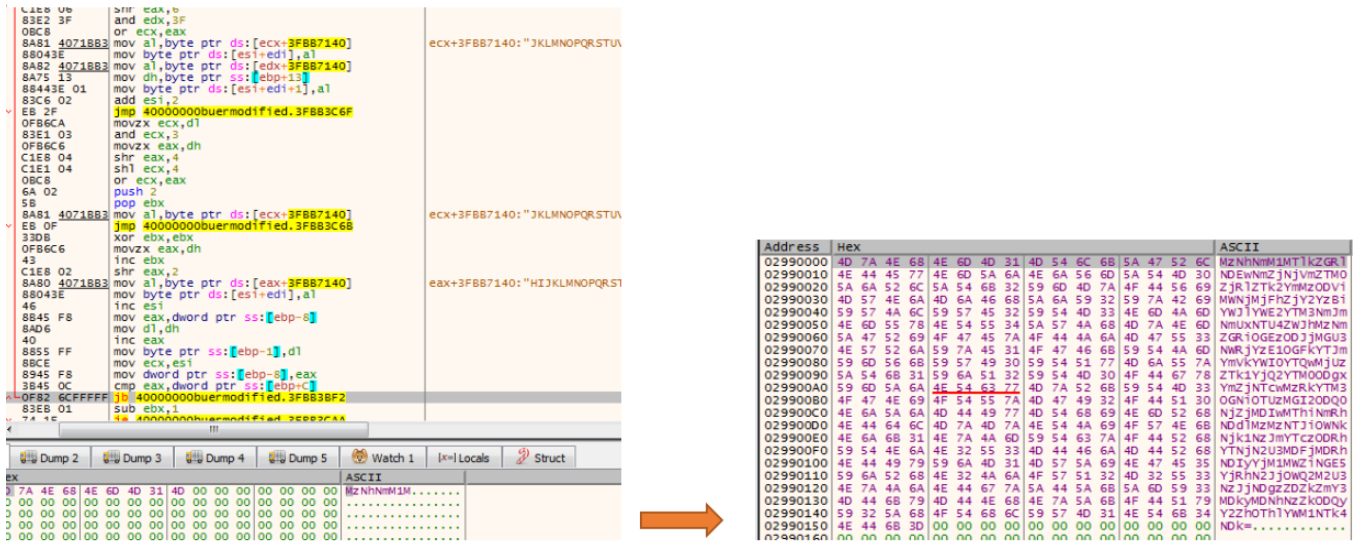


Figure 15.0 Converting to base64 string

Digging deeper into our analysis, we encountered InternetOpenA function to initialize a use of the WinINet functions. Then, it will try to open an http session to "verstudiosan[.]com" using InternetConnectW function.

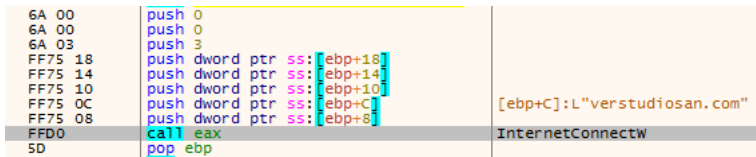


Figure 16.0 Opens an HTTP session

It has GET method to download additional malware and POST method to send the collected victim's machine information to the server:

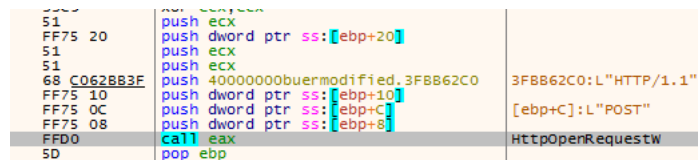


Figure 17.0 HTTP POST Request method

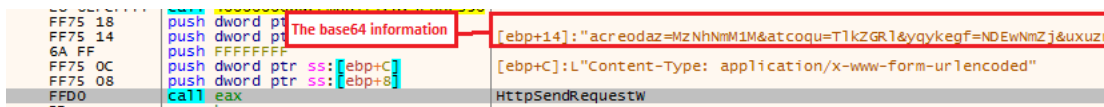


Figure 18.0 Sending the specified request

We searched the domain and found out that this domain was just recently created. We also learned that this domain is no longer reachable and possibly just used for malicious activity.

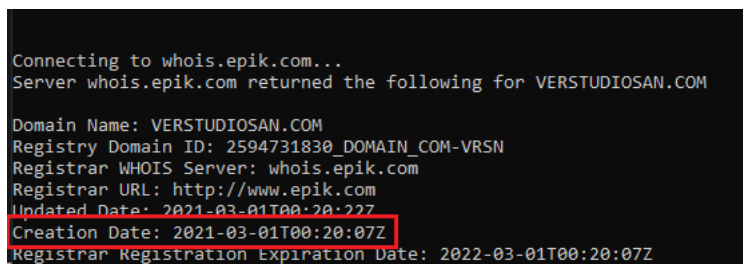


Figure 19.0 The recently created domain

```

586 194.595973      DNS      76 Standard query 0x2224 A verstudiosan.com
587 195.596485      DNS      76 Standard query 0x2224 A verstudiosan.com
• 588 196.531401      DNS      76 Standard query response 0x2224 Server failure A verstudiosan.com
589 196.670852      DNS      76 Standard query response 0x2224 Server failure A verstudiosan.com
590 196.670922      ICMP     104 Destination unreachable (Port unreachable)
591 197.742765      DNS      76 Standard query response 0x2224 Server failure A verstudiosan.com
592 197.742816      ICMP     104 Destination unreachable (Port unreachable)
593 203.560257      BROWSER 243 Local Master Announcement WIN-SPF5F5SH244. Workstation. Server. NT

```

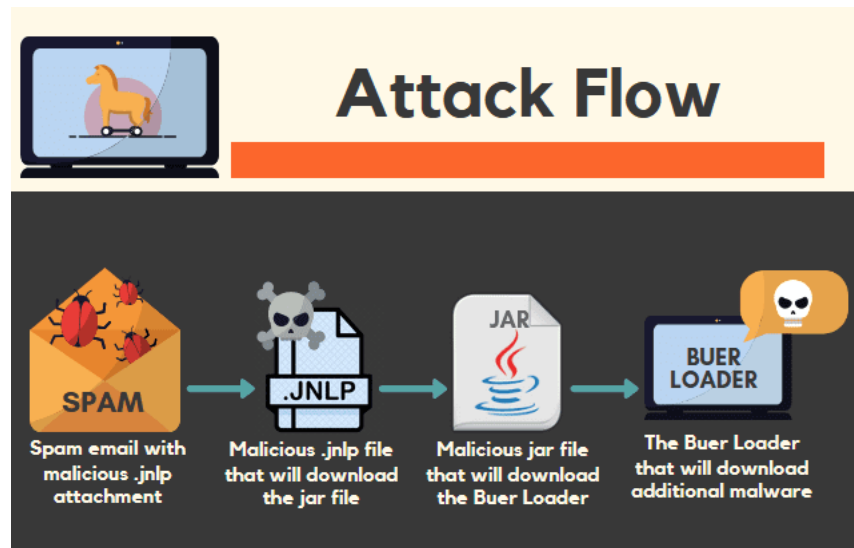
```

> Internet Control Message Protocol
v Domain Name System (response)
  > Transaction ID: 0x2224
  > Flags: 0x8182 Standard query response, Server failure
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  v Queries
    > verstudiosan.com: type A, class IN
    [Retransmitted response. Original response in: 588]

```

Figure 20.0 Unreachable server

Attack Flow



VIPRE detects and prevents this kind of malware and associated infections.

IOCs:

- The Spam Email
 - 66f13fa2c9e34705bbbc4645462188ca57c0fdc3a17418c96c0ed9371055f3bc
- JNLP File
 - 368b409080e9389b342e33a014cd7daf3fd984fdc2b0e5ecc8ac4d180759a1c4
- Jar File
 - 064fe7ef429f373d38813a05c9d2286a86337c1fc1b12c740b729f1f76de1811
- PE File
 - dbdc38dee1c9c9861a36cf6462dca55dcef6c1f128b2270efd99d4347568292c
- Malicious website
 - verstudiosan[.]com
 - hxxp://invoicsecure[.]net/documents

Analysis by #Farrallel