AlienSpy Payload Analysis | Proofpoint US

By August 14, 2015 Thoufique Haq

Published: 2015-08-15 · Archived: 2026-04-05 16:28:02 UTC

The media recently reported [1] on a potential targeted cyberattack on Alberto Nisman, an Argentine prosecutor who was found dead under mysterious circumstances. The file purportedly found on his phone was reported by the Argentine media to be named “Estrictamente Secreto y Confidencial.pdf.jar”, and as discovered and reported by the researcher Marquis-Boire, a file was indeed uploaded to VirusTotal [2] with the same name. The file was uploaded from Argentina and the upload is time stamped to “2015-05-29 18:48:24”. The file is JAR (Java Archive) file and is a variant of the AlienSpy Remote Access Trojan, or RAT.

AlienSpy RAT supports multiple platforms, and payloads can be crafted for Windows, Linux, Mac OS X and Android operating systems. However, contrary to popular reporting, the JAR file may not have been intended as a payload for the prosecutor’s Android phone. Android payloads are typically APKs (Android Application Package) or native binaries compiled for ARM processors. While technically possible, it is not trivial to get JAR payloads to run on an Android operation system without installing a Java emulation engine. Instead, it is more likely that the payload was intended to be loaded on a desktop environment but may have been inadvertently downloaded on the Android phone, possibly through email or another vector.

Proofpoint threat analysts obtained a copy of the purported payload in order to examine it more closely, and this post will follow the process of that analysis. There already exist several good analyses on the AlienSpyRAT [3] [4], and good work has also been done writing decoders and configuration extractors for it [5]. This post will therefore focus on aspects of AlienSpy that have not been previously examined in detail in these other sources.

Technical Analysis

AlienSpy payloads seen in the wild are typically obfuscated using Allatori, a commercial obfuscation product for Java that makes the code unreadable to prying eyes in order to evade easy detection. This particular payload was however packaged in an additional layer of obfuscation, which is not commonly seen in recent in-the-wild payloads for AlienSpy. The file structure of the top-level payload is shown in Figure 1.

```
 Length   Method    Size  Ratio   Date   Time   CRC-32    Name
--------  ------  ------- -----   ----   ----   ------    ----
     200  Defl:N      160  20%  12-01-14 11:47  17141395  META-INF/MANIFEST.MF
   62562  Defl:N    62433   0%  12-01-14 11:47  19088ea6  Favicon.ico
    4151  Stored     4151   0%  08-17-13 10:20  4d4745aa  Principal.class
--------          -------  ---                            -------
   66913            66744   0%                            3 files
```

Figure 1: Structure of top-level JAR payload

Note the large “Favicon.ico” file, not typically seen within JAR archives. When looking at the code it becomes clear that it is in fact an embedded JAR file that is loaded and executed in turn. The contents of the file are loaded using the

method getResourceAsStream(), saved in a temporary path and then launched using the command “java –jar payload.jar”.

```
private String getFav()
{
    ....
    return "Favicon.ico";
}

private InputStream asdasdasdasdaspdaksdpoakspodkaspokdaposkodpaksdpkp()
 {
    ....
    return getClass().getResourceAsStream(getFav());
}

... .
Runtime.getRuntime().exec(new String[] { "java", "-jar",
    dasndjansdjnasjdnaksdnasjdnaskjdnasjkdnasjdasd.getAbsolutePath() });
```

Figure 2: Code snippets showing JAR file being unpacked from Favicon.ico

The JAR file that is unpacked and loaded has a structure that is commonly seen with Allatori-obfuscated files. (Fig. 3) Allatori obfuscates class and variable names, making them difficult to read, and encrypts the original payload in a resource object.

There is also an embedded PDF file in the JAR archive called “Estrictamente Secreto y Confidencial.pdf” as seen in Figure 3. This is a decoy document that is launched in the foreground as the AlienSpy payload is executed, although the decoy document itself was found to be blank.

```
Archive:  0doc.jar
  Length      Date    Time    Name
 --------    ----    ----    ----
      136  12-01-14 11:22   META-INF/MANIFEST.MF
    43697  12-01-14 11:22   MANIFEST.MF
        8  12-01-14 11:22   ID
      562  12-01-14 11:22   plugins/Server.class
     7929  12-01-14 11:22   Main.class
     8134  12-01-14 12:28   Estrictamente Secreto y Confidencial.pdf
 --------                   -------
    60466                   6 files
```

Figure 3: Structure of embedded JAR file with Allatori obfuscation

A demo version of Allatori v5.1 is used in this instance. The payload is encrypted using an RC4 encryption scheme. (Fig. 4) The first clue that it is RC4 is the 256-byte array that is initialized with positional values 0-255 (0x00 – 0xFF). The array is then scrambled in the subsequent loop with the key stored in the variable “iiIIIIIiII4”. This 256-byte array is the substitution box or “Sbox” used in the RC4 encryption scheme.

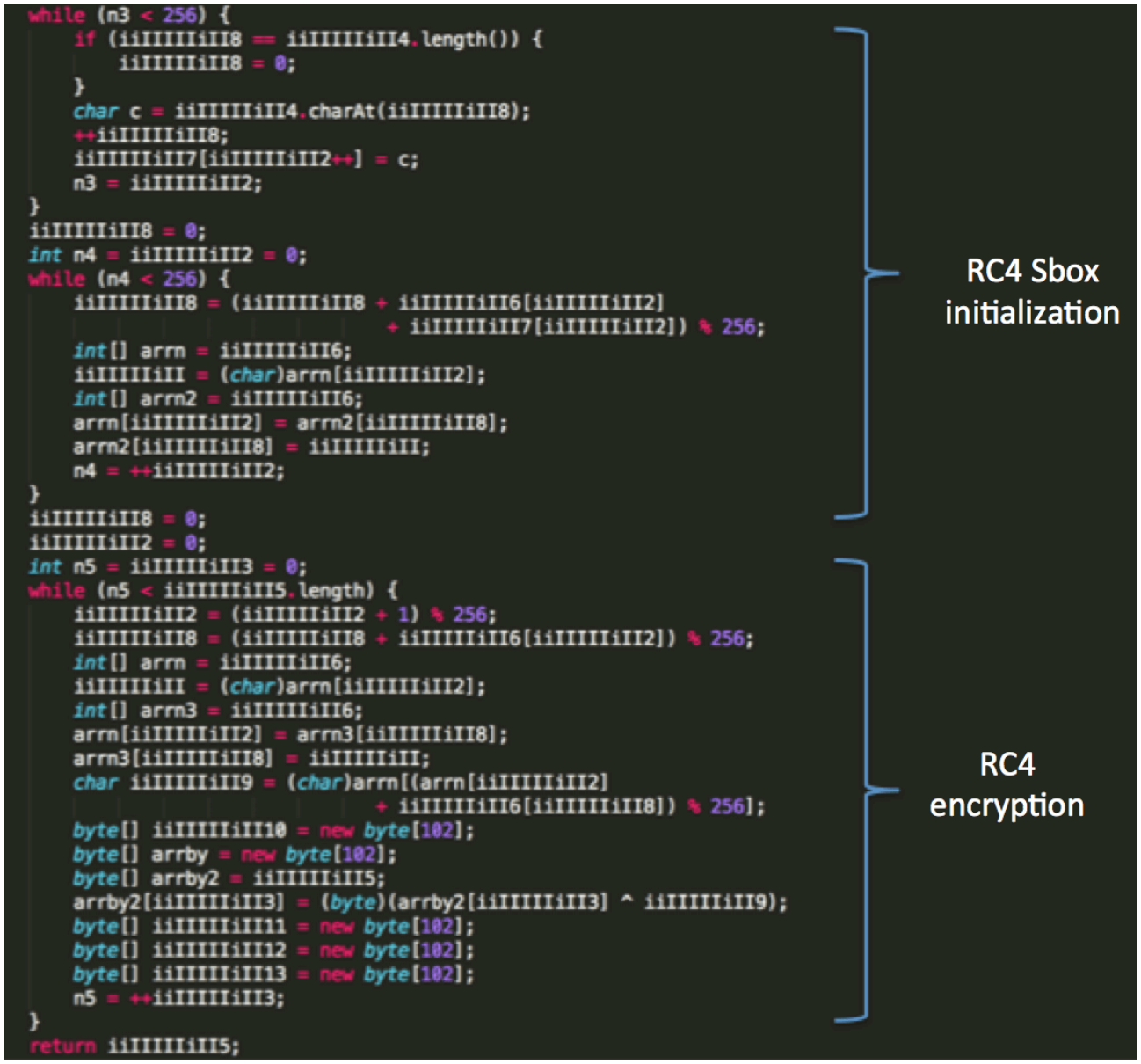


Figure 4: RC4 encryptions scheme used to decrypt the final AlienSpy payload

The RC4 key is constructed from a combination of a dynamic and a static string. The static string is hardcoded in the code with the value “H3SUW7E82IKQK2J2J2IISIS”. The dynamic portion of the string is a 8 byte value that is extracted from the resource section the object called “ID”. The dynamic portion of the key is randomly generated at build time of the payload. The combined RC4 in this instance was “oVs0Jp7kH3SUW7E82IKQK2J2J2IISIS”. We have also observed various other static keys being used with AlienSpy in samples seen in the wild.

Once decrypted the following file structure is observed in the final decrypted AlienSpy JAR payload, revealing a wide range of features. These features can be further extended through secondary plugins the attacker can choose to push down to the victim’s machine.

```
-rw----     2.0 fat      196 bX defN  1-Dec-14 11:22 META-INF/MANIFEST.MF
-rw----     2.0 fat     1622 bl defN  1-Dec-14 11:22 config.xml
-rw----     1.0 fat        0 b- stor 13-Nov-14 16:41 options/
-rw----     1.0 fat        0 b- stor 13-Nov-14 16:41 plugins/
-rw----     1.0 fat        0 b- stor 13-Nov-14 16:41 util/
-rw----     2.0 fat     6058 b- defN 13-Nov-14 16:41 Start.class
-rw----     2.0 fat     1625 b- defN 13-Nov-14 16:41 options/ActivePlugin.class
-rw----     2.0 fat     1336 b- defN 13-Nov-14 16:41 options/AlienSpy$1.class
-rw----     2.0 fat     4669 b- defN 13-Nov-14 16:41 options/AlienSpy.class
-rw----     2.0 fat     2162 b- defN 13-Nov-14 16:41 options/DownloadAndExecute.class
-rw----     2.0 fat     4544 b- defN 13-Nov-14 16:41 options/Informacion.class
-rw----     2.0 fat     6518 b- defN 13-Nov-14 16:41 options/Install.class
-rw----     2.0 fat      795 b- defN 13-Nov-14 16:41 options/MessageBox.class
-rw----     2.0 fat     2156 b- defN 13-Nov-14 16:41 options/OpenURL.class
-rw----     2.0 fat     3729 b- defN 13-Nov-14 16:41 options/RecibePlugin.class
-rw----     2.0 fat     4089 b- defN 13-Nov-14 16:41 options/RestartConnexion.class
-rw----     2.0 fat      695 b- defN 13-Nov-14 16:41 options/RunJarThread.class
-rw----     2.0 fat     5985 b- defN 13-Nov-14 16:41 options/Uninstall.class
-rw----     2.0 fat     2684 b- defN 13-Nov-14 16:41 options/Update_Offline.class
-rw----     2.0 fat     2405 b- defN 13-Nov-14 16:41 options/Update_Online.class
-rw----     2.0 fat     1039 b- defN 13-Nov-14 16:41 plugins/CargadorPlugins$1.class
-rw----     2.0 fat     3077 b- defN 13-Nov-14 16:41 plugins/CargadorPlugins.class
-rw----     2.0 fat     2318 b- defN 13-Nov-14 16:41 plugins/ModificadorClassPath.class
-rw----     2.0 fat      698 b- defN 13-Nov-14 16:41 plugins/PluginsTotales$1.class
-rw----     2.0 fat     1795 b- defN 13-Nov-14 16:41 plugins/PluginsTotales.class
-rw----     2.0 fat      500 b- defN 13-Nov-14 16:41 plugins/Server.class
-rw----     2.0 fat      339 b- defN 13-Nov-14 16:41 util/Config.class
-rw----     2.0 fat      624 b- defN 13-Nov-14 16:41 util/Control$1.class
-rw----     2.0 fat     3876 b- defN 13-Nov-14 16:41 util/Control.class
-rw----     2.0 fat     5656 b- defN 13-Nov-14 16:41 util/ForceRegedit.class
-rw----     2.0 fat     4117 b- defN 13-Nov-14 16:41 util/Utils.class
```

Figure 5: Structure of final decrypted payload

The AlienSpy RAT is very powerful in the hands of an attacker. Some of the key features supported by the RAT include:

- Collection of system information for fingerprinting and displaying on the attacker's controller dashboard
- File system, process and registry explorer with ability to view and modify
- Ability to run console commands
- Keylogging to capture user inputs
- Ability to download and execute secondary payloads
- Credential theft from various browser stores
- Ability to spy on victim through screenshots, webcam, microphone
- Ability to RDP (Remote Desktop) to infected clients
- Ability to mine various type of digital currency such as bitcoin, litecoin, dogecoin etc.

The config.xml file seen in the decrypted structure in Figure 5 contains the configuration of the RAT. The extraction configuration is shown below:

<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<!DOCTYPE properties SYSTEM "[http://java.sun.com/dtd/properties.dtd[">

<properties>

<comment>AlienSpy</comment>

<entry key="pluginfolder">cOzdAJCuee</entry>

<entry key="reconnetion_time">3000</entry>

<entry key="ps_hacker">false</entry>

<entry key="restore_system">false</entry>

<entry key="pluginfoldername">cOzdAJCuee</entry>

<entry key="dns">deyrep24.ddns.net</entry>

<entry key="install_time">3000</entry>

<entry key="port2">1040</entry>

<entry key="port1">1030</entry>

<entry key="taskmgr">false</entry>

<entry key="vmware">true</entry>

<entry key="jarname">documentos</entry>

<entry key="msconfig">false</entry>

<entry key="mutex">wMiSl1X1o423a2hh45Uifk8duasdf2S</entry>

<entry key="install">true</entry>

<entry key="instalar">true</entry>

<entry key="vbox">true</entry>

<entry key="password">ca19d6a81d35685b87547898c5e000a5fc9be554</entry>

<entry key="NAME">Localhost</entry>

<entry key="extensionname">jHs</entry>

<entry key="prefix">officce</entry>

<entry key="jarfoldername">0o86gb96</entry>

<entry key="uac">false</entry>

<entry key="win_defender">false</entry>

<entry key="connetion_time">3000</entry>

<entry key="folder">0o86gb96</entry>

<entry key="jar">documentos</entry>

<entry key="pluginextension">jHs</entry>

<entry key="registry">389032</entry>

<entry key="ps_explorer">false</entry>

<entry key="p2">1040</entry>

<entry key="p1">1030</entry>

<entry key="registryname">389032</entry>

<entry key="wireshark">false</entry>

<entry key="desktop">true</entry>

<entry key="nickname">officce</entry>

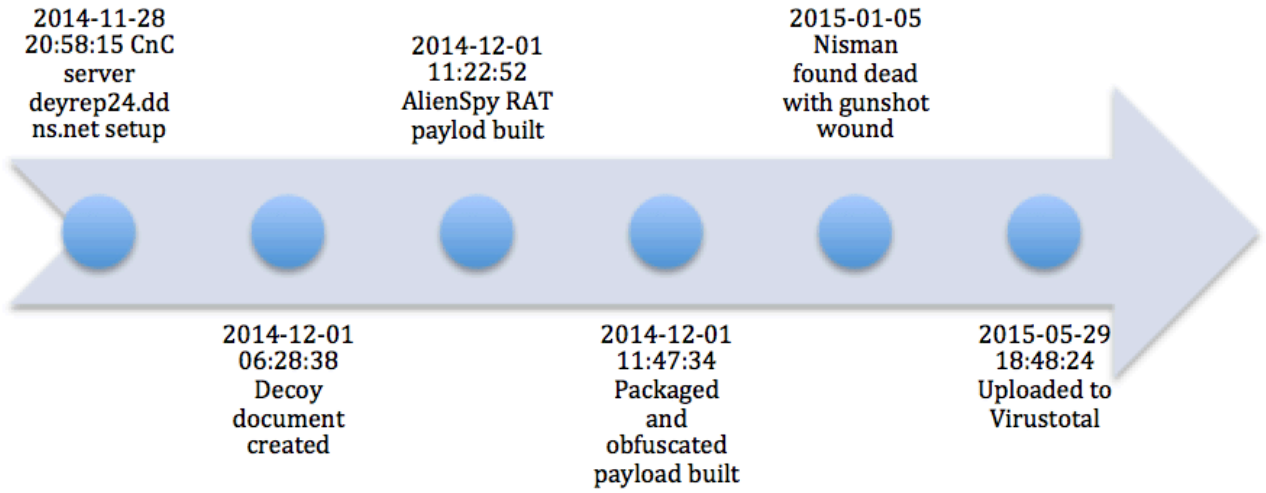
</properties>

The configuration stores various settings such as the remote server and port to connect to, install paths, VM and security tools detection, registry persistence location, and others. It is interesting to note some Spanish language strings in the settings, such as “documentos,” which hint at the origins of the attacker.

The AlienSpy RAT authenticates to the remote command and control server using a connection password. The connection password in this instance is listed in the configuration as the value “ca19d6a81d35685b87547898c5e000a5fc9be554”. This value is a SHA1 hash of the string “7854”. The significance of this password to the attacker is unknown but it may serve as good reference for future attack pivots since attackers often reuse settings by habit.

Timeline of events

This timeline of the analyzed sample is constructed based on the various time stamps observed on the files, with the caveat knowledgeable attackers are capable of altering digital timestamps.



Conclusion

Attacks in the digital world are often a consequence of ongoing events in the real world. It is unclear whether this attack payload has any relevance to death of Alberto Nisman, but the payload analyzed nonetheless shows how the immense capability that a malware payload provides. RATs such as AlienSpy constitute powerful surveillance tools that would enable them to observe and collect information on the communications and actions of adversaries, making them attractive to state actors and cybercriminals alike. As we continue to observe 'crossover' in the traditional targeting of state actors and cybercriminals, public and private organizations need to be on alert for phishing and other attacks designed to deliver RATs such as AlienSpy onto client systems.

Payload hash:

aa9aa05af8df2cc99eb936e2d17623a68abdbb60606bb097379457c4a3760116

References:

[1] http://motherboard.vice.com/read/malware-hunter-finds-spyware-used-against-dead-argentine-prosecutor

[2] https://www.virustotal.com/en/file/aa9aa05af8df2cc99eb936e2d17623a68abdbb60606bb097379457c4a3760116/analysis/

[3] http://blog.idiom.ca/2015/03/AlienSpy-java-rat-overview.html

[4] http://www.fidelissecurity.com/sites/default/files/FTA_1015_AlienSpy_FINAL.pdf

[5] https://github.com/idiom/IRScripts/blob/master/AlienSpy-decrypt-v2.py

---