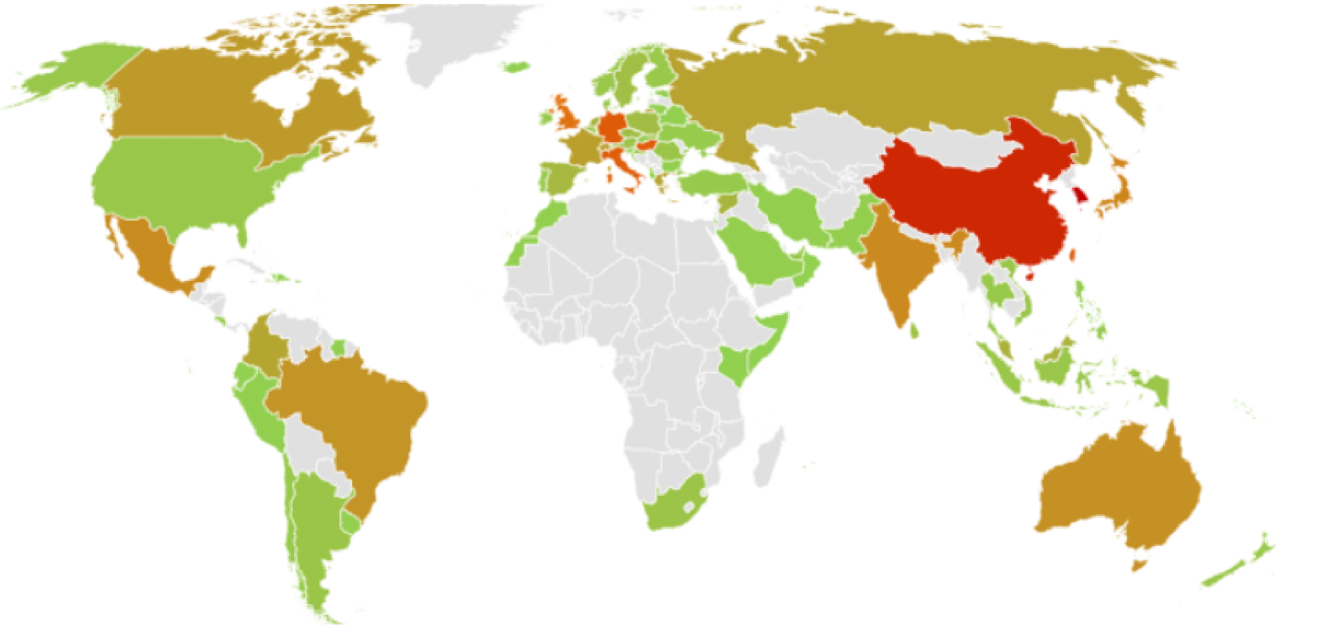
The Stealthy Email Stealer in the TA505 Arsenal

blog.yoroi.company/research/the-stealthy-email-stealer-in-the-ta505-arsenal/

May 16, 2019



05/16/2019

Introduction

During the last month our Threat Intelligence surveillance team spotted increasing evidence of an operation intensification against the Banking sector. In fact, many independent researchers pointed to a particular email attack wave probably related to the known TA505 hacking group, active since 2014 and focusing on *Retail* and *Banking* companies. The group is also known for some evasive techniques they put in place over time to avoid the security controls and penetrate corporate perimeters with several kinds of malware, for instance abusing the so called LOLBins (Living Off The Land Binaries), legit programs regularly used by victim, or also the abuse of valid cryptographically signed payloads.

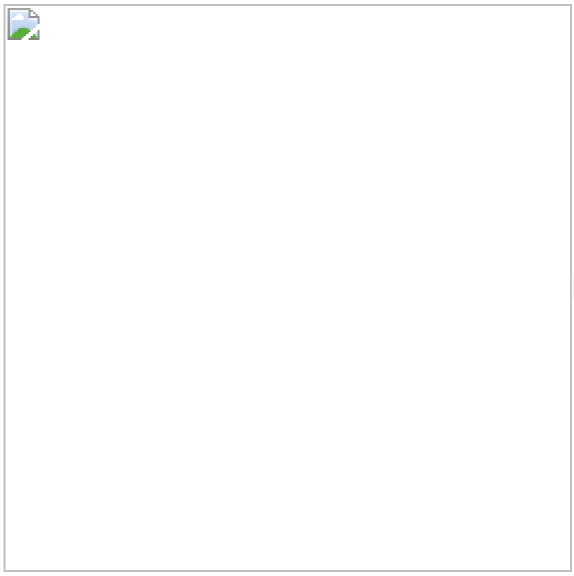


Figure 1. Attack campaign spotted in the wild.

Investigating and tracking their operations during April and May we detected an interesting tool was delivered through the victim machine. Just after the opening of malicious documents and the installation of FlawedAmmy RAT implants, the group used to deploy a particular credential stealing software, part of their arsenal, revealing details of their recent operation.

Figure 2. Attack campaign spotted in the wild.

Technical Analysis

The piece of malware under analysis were downloaded from “bullettruth[.com/out[.exe”, it was executed into the victim machines after the establishment of the infection.

Sha256	f3e8f68c31c86d431adea1633c875c32434a42aee5ed70af74af5c5e5aa58883
Threat	Custom Email Stealer
Brief Description	Executable of the email stealer
Ssdeep	12288:tlICpzmDFPJ+d7SQX5PsTrKjL43vNa77pu:XI+mDFx+d7vcrKv43X

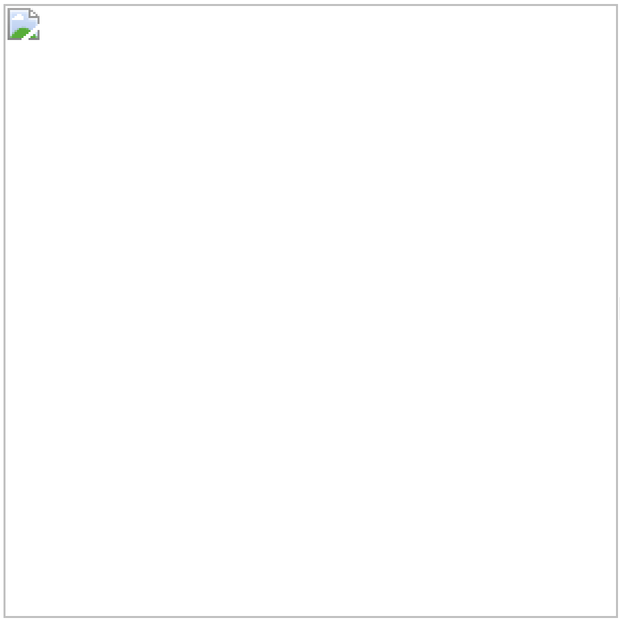


Figure 3: Malware Signature by SLON LTD

Firstly, we noticed this secondary component was well protected against antivirus detection, in fact the PE file was signed by Sectigo in the first half of May, one of the major Russian Certification Authority. Analyzing the trust chain we found the attackers were relying on cryptographic keys released to a UK company named **SLON LTD**. At this time, we have no evidence to hypothesize it could be victim of previous hacks or not.

Anyway, a static inspection of the binary revealed that the malware has a quite high entropy level, suggesting it may be packed.

Figure 4: Malware suspicious entropy level

Dynamically executing the malware, more information about its behaviour is revealed. The malicious executable is substantially an email stealer, in fact, the only purpose is to retrieve all the emails and passwords accounts present inside the victim machine. After executing the information gathering routine, the malware sends to its C2 all the retrieved emails and passwords:

Figure 5: HTTP POST communication

The interesting thing about the communication with the C2 is the fact that there is no encryption: the data harvested are sent to the C2 in JSON format. Investigating the attacker infrastructure we noticed interesting information such as the information of the stolen emails through our Digital Surveillance systems.

In order to retrieve more details about this Email Stealer, the analysis has moved into debugging and disassembling. As previously mentioned, the malware sample is heavily obfuscated and packed. However, by letting the malware execute itself within a debugger, we were able to extract the unpacked payload of the malware.

Figure 6: Static information about the packed sample (on the left) and the unpacked one (on the right)

As shown by the above figure, we notice a peculiarity of these two components: while the packed sample is compiled in *Microsoft Visual C++ version 6.0*, the unpacked one is compiled in *Microsoft Visual C++ version 8*. At this point, we deepen the analysis on the extracted payload. However, we are not able to execute it, because it always references many memory addresses of the original one. So, we carry on static analysis on the extracted sample.

As previously described, the malware's principal purpose is to iterate through the filesystem looking for email accounts.. The first step is to check whether the "*outlook.exe*" process is running and, in this case it kills the process.The malware iterate through user processes with *Process32FirstW* API and then kill it with *TerminateProcess:*

Figure 7: Outlook process search routine
The extracted payload does not present any type of code obfuscation of other types. In fact the C2 server and the path is not encoded:

Figure 8: C2 connection routine
The last routine being analyzed is the credential harvesting inside the entire filesystem.

Apart from the routine that searches for the email account registered in Outlook and Thunderbird clients (as shown in Figure 7), there is another one which scans the filesystem looking for hardcoded extensions, then, if one of them is found, a reference to the found file is conserved inside the %TEMP% directory. At this point, all the gathered email accounts are sent to the server and then erasing all traces of itself from the infected machine, in fact, the malware creates a simple batch script which delete itself and all the tracks of infection.

Figure 9: Autodeletion batch script

Analysis of Exposed Emails

In this paragraph are shown some statistics about the harvested emails in the attack campaign, recovered during surveillance and hunting operations. So we decided to create a graph in which sort the most frequent TLD occurrences of all the stolen data.

Figure 10: Distribution of TLD
As seen in the graph above, the most frequent TLD is *.com* with 193.194 occurrences, following *.kr* with 102.025 occurrences, *.cn* with *26.160* occurrences*, it* with *6.317* occurrences and so on. To better visualize the macro-locations involved in this exposure we built a *heatmap* showing the geographical distribution of the TOP 100 countries referenced in the TLDs.

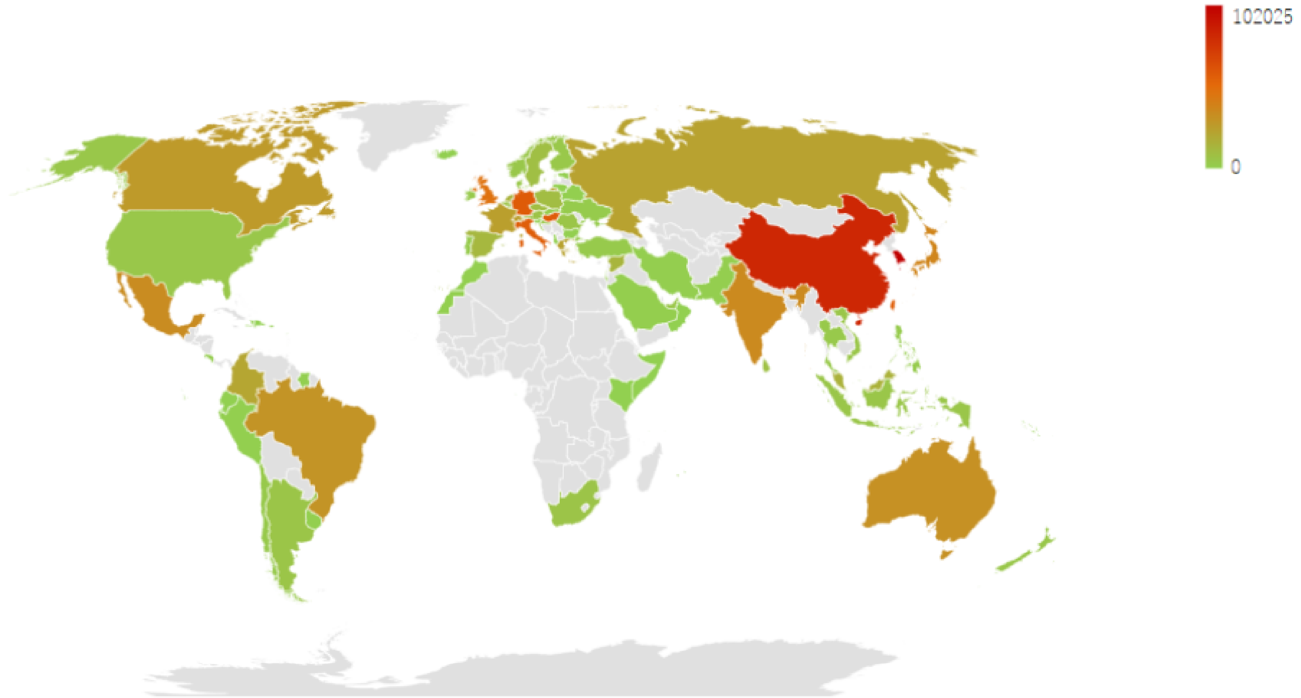


Figure 11: Geolocation of emails TLD exposure

The heatmap shows the less-affected countries with a greenish color, on the contrary, the most-affected ones tend to an orange or red-tinged color. The first thing that emerges from these 2 distributions is that this specific threat seems not to be targeted, in fact, the diffusion is almost global with some red or orange zones in UK, Italy, Republic of Korea, China, Germany, Hungary, Taiwan, Japan, India and Mexico. All these countries exceeded the thousand occurrences.

Conclusion

Nowadays, the email accounts are an effective source of revenue for the cyber criminals. In fact all these information can be used to spread other malware through phishing campaigns, to perform BEC attacks (Business Email Compromise) and also to try credential stuffing attacks.

Evan a simple Info-Stealer malware like this one could be a dangerous threat, especially if used by organized groups in conjunction with other malware implants. In fact, as reported by the independent researcher Germán Fernández Bacian too, this Email Stealer has been recently used by the infamous TA505 hacking group. This link means, with good confidence, the exposed data, full email accounts in some cases and email contacts in general, are now available to a cyber-criminal group who launched targeted attacks against Banks and Retail industries in the near past.

Indicators of Compromise

- Dropurl:
  bullettruth[.com/out[.exe

- C2:
  - nettubex[.top/es/es[.php
  - 178.48.154.38
  - 5.253.53.236
  - 87.241.136.1
  - 197.255.225.249
  - 95.140.195.178
  - 186.74.208.84
  - 86.61.75.99
  - 86.101.230.109
  - 89.47.94.113
  - 130.204.181.90
  - 78.90.243.124
- Hash:
  - 104dae7457c10b7fe6c42a335f2a57ff708ff20d70597fbaa5fe0083c1c628c7
  - e4b40cba02dc1de1a1c2ed2001d39a87c476c11ca08f09a80fd3f1fbaae0daeb
  - f3e8f68c31c86d431adea1633c875c32434a42aee5ed70af74af5c5e5aa58883
  - 899bfac53c3439a7ea68f9a5bbff2733ebf7b9158f18ef5d03360a09b18b5e0d

Yara Rules

```
import "pe"
rule EmailStealer_201905 {
meta:
        description = "Yara rule for EmailStealer"
        author = "Cybaze - Yoroi ZLab"
        last_updated = "2019-05-14"
        tlp = "white"
        category = "informational"
strings:
        $a1 = { 80 F2 F3 00 56 53 A7 }
        $a2 = { 4D 26 9A 00 56 4B AC 55 }
        $a3 = { 1C 4A 77 00 00 89 B4 B7 }


condition:
        uint16(0) == 0x5A4D and pe.number_of_sections == 3 and all of them
}
```

Searched Extensions

```
.msf; .dat; .pst; .ost; .asp; .cdd; .cpp; .doc; .docm; .docx; .dot; .dotm; .dotx;
.epub; .fb2; .gpx; .ibooks; .indd; .kdc; .key; .kml; .mdb; .mdf; .mobi; .mso; .ods;
.odt; .one; .oxps; .pages; .pdf; .pkg; .pl; .pot; .potm; .potx; .pps; .ppsm; .ppsx;
.ppt; .pptm; .pptx; .ps; .pub; .rtf; .sdf; .sgml; .sldm; .snb; .wpd; .wps; .xar;
.xlr; .xls; .xlsb; .xlsm; .xlsx; .xlt; .xltm; .xltx; .xps; .3dm; .aspx; .cer; .cfm;
.chm; .crdownload; .csr; .css; .download; .eml; .flv; .htaccess; .htm; .html; .jnlp;
.js; .jsp; .magnet; .mht; .mhtm; .mhtml; .msg; .php; .prf; .rss; .srt; .stl; .swf;
.torrent; .url; .vcf; .webarchive; .webloc; .xhtml; .xul; .asf; .asm; .cgi; .class;
.cs; .dtd; .fla; .ged; .gv; .icl; .java; .jse; .json; .lua; .mb; .mod; .msp; .obj;
.po; .ps1; .py; .sh; .sln; .so; .sql; .ts; .vbe; .vbs; .vc4; .vcproj; .vcxproj; .wsc;
.xcodeproj; .xsd; .apt; .err; .log; .pwi; .sub; .ttf; .tex; .text; .txt; .accdb; .b2;
.crypt; .crypt5; .crypt6; .crypt7; .crypt8; .crypt12; .db; .dbf; .dbx; .sis; .awb;
.bin; .cdi; .cdr; .csv; .eap; .efx; .gam; .gbr; .gtp; .mpp; .msc; .mts; .otf; .nbk;
.nbp; .ndb; .prj; .rtp; .sav; .scppy; .tax2010; .tbl; .tmp; .vcd; .xml; .xsl; .xslt;
.bak; .dmp; .gho; .ghs; .v2i; .zip; .asx; .iff; .inf; .temp; .ai; .aif; .amr; .apk;
.bp1; .ccd; .cdw; .dds; .dmg; .dxf; .ext; .ics; .ini; .m4p; .max; .md0; .mng; .mp3;
.mpa; .msu; .nrg; .pak; .part; .pkpass; .psd; .rnd; .rom; .spl; .swb; .svg; .xla;
.application; .appref; .cfg; .conf; .config; .cpl; .cue; .deskthemepack; .diagcfg;
.ds_store; .iso; .pdi; .plist; .reg; .scr; .theme; .themepack; .thm
```

This blog post was authored by Luigi Martire, Davide Testa, Antonio Pirozzi and Luca Mella of Cybaze-Yoroi Z-LAB