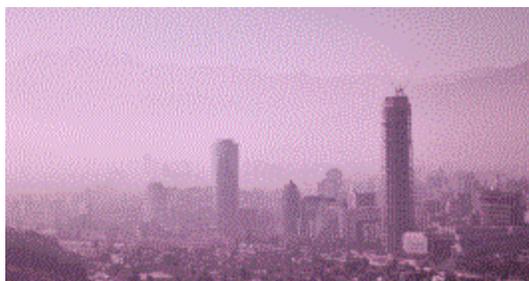


TDL4 - Purple Haze (Pihar) Variant - sample and analysis

 contagiodump.blogspot.com/2012/02/purple-haze-bootkit.html



Lately things just don't seem the same
Actin' funny, but I don't know why
'Scuse me..... while I kiss the sky
Jimi Hendrix "Purple Haze"

I recently ran into an interesting piece of malware that was downloaded on a victim's computer. I thought it was TDL/TDSS or maybe a new version of it as it had some components as TDL4 bootkit with a functionality of a mass scale PPC (pay-per-click) fraud. TDL had this functionality too and it is most likely spread by the same Russian-speaking gangs using the Blackhole exploit kit. It did not have the same type of config file that you may find in TDL4 (and first I could not find it at all). I call it "Purple Haze" thanks to the strings found in the code.

I shared it with Alexander Matrosov from ESET. He and Eugene Rodionov analyzed it and posted an article on the ESET blog: ["TDL4 reloaded: Purple Haze all in my brain"](#) (edited by [David Harley](#))

Eset also updated the removal tool for this variant - direct download link: [OlmarikTDL4 remover](#)

Distribution

The exploit host is featured on CleanMX . The domain was repossessed by GoDaddy after January 24, 2012 by but you can see some of the URLs. Infection happened via Blackhole exploit kit

95.211.115.228

General File Information

File: w.php.exe

Size: 130560

MD5: A1B3E59AE17BA6F940AF86485E5907

Download



[Download purplehazetdl.zip as a password protected archive \(contact me if you need the password\)](#)

[Download pcap BIN_purplehaze-pihar-A1B3E59AE17BA6F940AF86485E5907-2012-02.zip \(235MB\)](#)

Automatic scans

Original scan was only 2/43 but it is better now. It gets detected as a generic trojan or rootkit or as TDL/TDSS/Alureon.

Virustotal

SHA256: 9746b4f684b9d7d346ff131cd024e68d1b06e1b81571ce6d3c5067f0829d7932

SHA1: 6d07cf72201234a07ab57fb3fc00b9e5a0b3678e

MD5: a1b3e59ae17ba6f940afaf86485e5907

File size: 127.5 KB (130560 bytes)

File name: w.php.exe

File type: Win32 EXE

Detection ratio: 24 / 43

Analysis date: 2012-02-02 06:50:05 UTC (1 minute ago)

AntiVir TR/Alureon.FK.93 20120201

Avast Win32:Rootkit-gen [Rtk] 20120202

BitDefender Trojan.Generic.7154539 20120202

Comodo TrojWare.Win32.Trojan.Agent.Gen 20120202

DrWeb BackDoor.Tdss.5231 20120202

Emsisoft Trojan.Win32.FakeAV!IK 20120202

eSafe Win32.Rorpian.C 20120130

F-Secure Trojan.Generic.7154539 20120202
Fortinet W32/Rorpian.C!tr 20120202
GData Trojan.Generic.7154539 20120202
Ikarus Trojan.Win32.FakeAV 20120202
Kaspersky Trojan.Win32.FakeAV.kpsj 20120202 (TDSS Killer detects it as Pihar.b)
McAfee-GW-Edition Artemis!A1B3E59AE17B 20120202
Microsoft Trojan:Win32/Alureon.FK 20120202
NOD32 Win32/Olmarik.AYD 20120202
Norman W32/Troj_Generic.LPAP 20120201
Sophos Mal/Generic-L 20120202
TrendMicro-HouseCall TROJ_SPNR.16AQ12 20120202
VBA32 - 20120131
VIPRE Trojan.Win32.Generic!BT 20120202

Desription

You can read more detailed binary analysis on the ESET blog (Feb.2 2012) : ["TDL4 reloaded: Purple Haze all in my brain"](#)

Update. Feb 2, 2012

I heard today it is a recent but known variant detected by Kaspersky as "Pihar", which is supposedly a member of the TDL/TDSS/Olmarik/Alureon/ - Maxss family that does not encrypt the hidden container. I have to say I saw that Kaspersky detected it as Pihar.b via TDSS Killer (the dropper is detected as FakeAV) but it was a totally different name and I could not find any explanation of how Pihar is different from TDL4 - whether it is a misdetection, a different rootkit, some generic signature name, or a different variant of TDL. With the number of malware variants these days in the wild, it does not surprise me that it was known to them but there was no analysis posted (or I did not find it). I hope this analysis and the work done by ESET will make the family description more complete. TDSS Killer also removes it.

It is a kernel mode rootkit compatible with x86 and x64 Windows. It uses dll injection ph.dll and phx.dll (for x64). It creates a hidden VFS to store all the data.

The list of hidden system files:

1. Phdata
[PurpleHaze]
pn=161
all=ph.dll
allx=phx.dll
wait=3600
2. phm (original master boot record)
3. ph.dll (payload dll for x86)
4. phx.dll (payload dll for x64)


```

10006485 000000 Align 4
10006488 SSZ10006488_GlobalUserOffline:
10006488 476C6F62616C55736572+ db 'GlobalUserOffline',0
1000649A 0000 Align 4
1000649C SSZ1000649C_CertificateRevocation:
1000649C 43657274696669636174+ db 'CertificateRevocation',0
100064B2 0000 Align 4
100064B4 SSZ100064B4_WarnonBadCertRecving:
100064B4 5761726E6F6E42616443+ db 'WarnonBadCertRecving',0
100064C9 000000 Align 4
100064CC SSZ100064CC_WarnOnPost:
100064CC 5761726E4F6E506F7374+ db 'WarnOnPost',0
100064D7 00 Align 4
100064D8 SSZ100064D8_WarnOnPostRedirect:
100064D8 5761726E4F6E506F7374+ db 'WarnOnPostRedirect',0
100064EB 00 Align 4
100064EC SSZ100064EC_WarnonZoneCrossing:
100064EC 5761726E6F6E5A6F6E65+ db 'WarnonZoneCrossing',0
100064FF 00 Align 4
10006500 SSZ10006500_EnableHttp1_1:
10006500 456E61626C6548747470+ db 'EnableHttp1_1',0
1000650E 0000 Align 4
10006510 SSZ10006510_MaxHttpRedirects:
10006510 4D617848747470526564+ db 'MaxHttpRedirects',0
10006521 000000 Align 4
10006524 SSZ10006524_SecuritySafe:
10006524 53656375726974795361+ db 'SecuritySafe',0
10006531 000000 Align 4
10006534

```

Change IE settings

Traffic

Pay-per-click fraud generates significant revenue for the botnet owners. [The 'Advertising Botnet'](#) article from Securelist explains the click fraud scheme in great detail.



"Advertising Botnet" by Securelist

C&C check-in upon install

1 22:17:47.765551	172.29.0.110	75.75.75.75	DNS	standard query A host.google.com
2 22:17:47.284888	75.75.75.75	172.29.0.110	DNS	standard query response A 143.156.16.158
3 22:17:47.285174	172.29.0.110	143.156.16.158	TCP	dka > http [SYN] seq=0 win=64240 len=0 MSS=1460 SACK_PERM=1
4 22:17:47.423618	143.156.16.158	172.29.0.110	TCP	http > dka [SYN, ACK] seq=0 Ack=1 win=65535 len=0 MSS=1460 SACK_PERM=1
5 22:17:47.423657	172.29.0.110	143.156.16.158	TCP	dka > http [ACK] seq=1 Ack=1 win=64240 len=0
6 22:17:47.423980	172.29.0.110	143.156.16.158	HTTP	GET /?0020C14A73AGPMJ0m05G019219V01fjyflEhUCDw7X117bqpxJ75uMAG67mJd0wKlww HTTP/1.0
7 22:17:47.565273	143.156.16.158	172.29.0.110	TCP	http > dka [ACK] seq=1 Ack=101 win=65535 len=0
8 22:17:47.572307	143.156.16.158	172.29.0.110	TCP	[TCP segment of a reasssembled PDU]
9 22:17:47.572583	143.156.16.158	172.29.0.110	HTTP	HTTP/1.1 200 OK (text/html)
10 22:17:47.572627	172.29.0.110	143.156.16.158	TCP	dka > http [ACK] seq=101 Ack=181 win=66258 len=0
11 22:17:47.572817	172.29.0.110	143.156.16.158	TCP	dka > http [FIN, ACK] seq=101 Ack=181 win=66258 len=0
12 22:17:47.731760	143.156.16.158	172.29.0.110	TCP	http > dka [ACK] seq=184 Ack=102 win=65534 len=0
13 22:17:48.802793	172.29.0.110	143.156.16.158	TCP	prst > http [SYN] seq=0 win=64240 len=0 MSS=1460 SACK_PERM=1
14 22:17:48.745384	143.156.16.158	172.29.0.110	TCP	http > prst [SYN, ACK] Seq=0 Ack=1 win=65535 len=0 MSS=1460 SACK_PERM=1
15 22:17:48.745434	172.29.0.110	143.156.16.158	TCP	prst > http [ACK] Seq=1 Ack=1 win=64240 len=0
16 22:17:48.745981	172.29.0.110	143.156.16.158	HTTP	GET /?0020C14A73AGPMJ0m05G019219V01fjyflEhUCDw7X117bqpxJ75uMAG67mJd0wKlww HTTP/1.0
17 22:17:48.952071	143.156.16.158	172.29.0.110	TCP	http > prst [ACK] Seq=1 Ack=141 win=65535 len=0
18 22:17:48.952681	143.156.16.158	172.29.0.110	TCP	[TCP segment of a reasssembled PDU]
19 22:17:48.953096	143.156.16.158	172.29.0.110	HTTP	HTTP/1.1 200 OK (text/html)
20 22:17:48.953149	172.29.0.110	143.156.16.158	TCP	prst > http [ACK] Seq=141 Ack=232 win=63990 len=0
21 22:17:48.953491	172.29.0.110	143.156.16.158	TCP	prst > http [FIN, ACK] Seq=141 Ack=232 win=63990 len=0
22 22:17:48.953908	172.29.0.110	143.156.16.158	TCP	0ss1api > http [SYN] seq=0 win=64250 len=0 MSS=1460 SACK_PERM=1
23 22:17:49.054824	143.156.16.158	172.29.0.110	TCP	http > 0ss1api [ACK] Seq=252 Ack=142 win=65534 len=0
24 22:17:49.071658	143.156.16.158	172.29.0.110	TCP	http > 0ss1api [SYN, ACK] seq=0 Ack=1 win=65535 len=0 MSS=1460 SACK_PERM=1
25 22:17:49.071890	172.29.0.110	143.156.16.158	TCP	0ss1api > http [ACK] Seq=1 Ack=3 win=64240 len=0
26 22:17:49.072037	172.29.0.110	143.156.16.158	HTTP	GET /?0020C14A73AGPMJ0m05G019219V01fjyflEhUCDw7X117bqpxJ75uMAG67mJd0wKlww HTTP/1.0
27 22:17:49.207190	143.156.16.158	172.29.0.110	TCP	http > 0ss1api [ACK] Seq=1 Ack=97 win=65535 len=0
28 22:17:49.207512	143.156.16.158	172.29.0.110	HTTP	HTTP/1.1 200 OK (application/octet-stream)
29 22:17:49.207547	172.29.0.110	143.156.16.158	TCP	0ss1api > http [ACK] Seq=97 Ack=373 win=63888 len=0
30 22:17:49.207903	172.29.0.110	143.156.16.158	TCP	0ss1api > http [FIN, ACK] Seq=97 Ack=373 win=63888 len=0
31 22:17:49.344513	143.156.16.158	172.29.0.110	TCP	http > 0ss1api [ACK] Seq=373 Ack=98 win=65534 len=0
32 22:17:54.799346	172.29.0.110	172.29.0.255	4000SER	Local Master: error:cannot XP01-831-05C2D, workstation, Server, NT workstation, Potential Browser, Master Browser
33 22:18:15.142226	172.29.0.1	172.29.0.110	TCP	funcrpt > 4000 [RST] Seq=1 win=0 len=0
34 22:20:37.298192	172.29.0.110	75.75.75.75	DNS	standard query A x-web.in
35 22:20:37.331255	75.75.75.75	172.29.0.110	DNS	standard query response A 178.238.233.156
36 22:20:37.331642	172.29.0.110	178.238.233.156	TCP	de11perapps > http [SYN] seq=0 win=64240 len=0 MSS=1460 SACK_PERM=1
37 22:20:37.451434	178.238.233.156	172.29.0.110	TCP	http > de11perapps [SYN, ACK] Seq=0 Ack=1 win=65535 len=0 MSS=1460 SACK_PERM=1
38 22:20:37.451473	172.29.0.110	178.238.233.156	TCP	de11perapps > http [ACK] Seq=1 Ack=1 win=64240 len=0
39 22:20:37.453779	172.29.0.110	178.238.233.156	HTTP	GET /?2486642f020c14a73agpmj0m05g019219v01fjyflEhUCDw7X117bqpxJ75uMAG67mJd0wKlww HTTP/1.0
40 22:20:37.575616	178.238.233.156	172.29.0.110	TCP	http > de11perapps [ACK] Seq=1 Ack=218 win=65535 len=0
41 22:20:37.576936	172.29.0.110	172.29.0.255	4000SER	Domain/workgroup: error:cannot MSHOME, NT workstation, Domain Erum
42 22:20:39.806382	178.238.233.156	172.29.0.110	HTTP	HTTP/1.1 200 OK (text/html)
43 22:20:39.806418	178.238.233.156	172.29.0.110	TCP	http > de11perapps [FIN, ACK] Seq=1309 Ack=238 win=65535 len=0
44 22:20:39.806494	172.29.0.110	178.238.233.156	TCP	de11perapps > http [ACK] Seq=238 Ack=1370 win=62872 len=0
45 22:20:39.807276	172.29.0.110	178.238.233.156	TCP	de11perapps > http [FIN, ACK] Seq=238 Ack=1370 win=62872 len=0
46 22:20:39.852872	172.29.0.110	178.238.233.156	TCP	apc > http [SYN] seq=0 win=64240 len=0 MSS=1460 SACK_PERM=1
47 22:20:39.934428	178.238.233.156	172.29.0.110	TCP	http > de11perapps [ACK] Seq=1370 Ack=239 win=65534 len=0
48 22:20:39.931009	178.238.233.156	172.29.0.110	TCP	http > apc [SYN, ACK] Seq=0 Ack=1 win=65535 len=0 MSS=1460 SACK_PERM=1
49 22:20:39.931681	172.29.0.110	178.238.233.156	TCP	apc > http [ACK] Seq=1 Ack=1 win=64240 len=0
50 22:20:39.932079	172.29.0.110	178.238.233.156	HTTP	GET /?2486642f020c14a73agpmj0m05g019219v01fjyflEhUCDw7X117bqpxJ75uMAG67mJd0wKlww HTTP/1.0
51 22:20:39.931511	172.29.0.110	172.29.0.110	HTTP	[TCP reset: seq=1370] Seq=1370 win=0 len=0 MSS=1460 SACK_PERM=1
52 22:20:35.931592	172.29.0.110	75.75.75.75	TCP	http > apc [ACK] Seq=1 Ack=218 win=65535 len=0
53 22:20:35.933089	172.29.0.110	75.75.75.75	DNS	standard query A @webbrowser.hud1.va.comcast.net
54 22:20:35.933984	172.29.0.110	75.75.75.75	DNS	standard query A @discoverfindsearch.net
55 22:20:35.948686	75.75.75.75	172.29.0.110	DNS	standard query response A 233.174.149.74
56 22:20:35.949082	172.29.0.110	233.174.149.74	TCP	opnpr > http [SYN] seq=0 win=64240 len=0 MSS=1460 SACK_PERM=1
57 22:20:35.949638	75.75.75.75	172.29.0.110	DNS	standard query response A no such name
58 22:20:35.954902	172.29.0.110	75.75.75.75	DNS	standard query A @webbrowser.hud1.va.comcast.net
59 22:20:35.969918	233.174.149.74	172.29.0.110	TCP	http > opnpr [SYN, ACK] Seq=0 Ack=1 win=5840 len=0 MSS=1460
60 22:20:35.969944	172.29.0.110	233.174.149.74	TCP	opnpr > http [ACK] Seq=1 Ack=1 win=64240 len=0

The bot generates high volume traffic to thousands of websites with ads, sites serving as referrers, as well as pages filled with ad links (over 800 sessions a minute) for approximately 2 hours and then stops. Most serious advertising companies easily detect large clicks from the same ip and block it. The botnet owners limit clicks to just a few and compensate it by programming the bot to click on thousands of ads.



Click to enlarge. 11 hours of traffic monitoring. 2 hour spike following the infection.

Traffic capture - Using fake referrer (serch-direct.com) and passing fake search strings to the C&C, which responds with iframe redirect to the ad link.

DOMAINS:

hosted-by.leaseweb.com

WhoisGuard

WhoisGuard Protected ()

Fax:

11400 W. Olympic Blvd. Suite 200

Los Angeles, CA 90064

United States

IPs:

Private Customer

Private Residence

Bryansk

241000

Russian Federation

In some cases, legitimate "traffic quality" providers were used as referrers, such as ezanga.com

```
NetWitness Reconstruction for session ID: 31 ( Source 172.29.0.116 : 1286, Target 69.31.72.136 : 80 )
Time 1/30/2012 22:20:39 to 1/30/2012 22:20:41 Packet Size 2,490 bytes Payload Size 1,814 bytes
Protocol 2048 (6:80) Flags Keep Assembled AppMeta NetworkMeta Packet Count 12

REQUEST
GET /p?c1=2&c2=8287123&c4=www.ezanga.com/search/web.php?cov=2.0&cj=1 HTTP/1.0
Accept: */*
Referer: http://1791264036.pub.ezanga.com/rv2.php?c7f1d1341e0d1b8caf92e68b17ec62e
3f31fad8d95&q=credit+counseling+and+debt+management
Accept-Language: en-US
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; InfoPath.1)
Host: b.scorecardresearch.com
Connection: Keep-Alive

RESPONSE
HTTP/1.0 302 Moved Temporarily
Content-length: 0
Location: http://b.scorecardresearch.com/p?c1=2&c2=8287123&c4=www.ezanga.com/search/web.php?cov=2.0&cj=1
Date: Tue, 31 Jan 2012 03:20:40 GMT
Connection: keep-alive
Set-Cookie: UID=lad1e12-69.31.72.136-1327980040; expires=Mon, 20-Jan-2014 03:20:40 GMT; path=/; domain=.scorecardresearch.com
Set-Cookie: UIDB=1327980040; expires=Mon, 20-Jan-2014 03:20:40 GMT; path=/; domain=.scorecardresearch.com
P3P: policyref="/w3c/pp.xml", CP="NOI DSP COR NID ORN IND COM STA OTC"
Pragma: no-cache
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Cache-Control: private, no-cache, no-cache=Set-Cookie, no-store, proxy-revalidate

REQUEST
GET /p?c1=2&c2=8287123&c4=www.ezanga.com/search/web.php?cov=2.0&cj=1 HTTP/1.0
Accept: */*
Referer: http://1791264036.pub.ezanga.com/rv2.php?c7f1d1341e0d1b8caf92e68b17ec62e
3f31fad8d95&q=credit+counseling+and+debt+management
Accept-Language: en-US
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1; InfoPath.1)
Host: b.scorecardresearch.com
Connection: Keep-Alive
Cookie: UID=lad1e12-69.31.72.136-1327980040; UIDB=1327980040

RESPONSE
HTTP/1.0 200 OK
Content-length: 43
Content-Type: image/gif
Pragma: no-cache
Date: Tue, 31 Jan 2012 03:20:40 GMT
Connection: keep-alive
Expires: Mon, 01 Jan 1990 00:00:00 GMT
Cache-Control: private, no-cache, no-cache=Set-Cookie, no-store, proxy-revalidate

GIF89a1,0:
```

The list of hosts involved (if you think you might be a PPC fraud victim, see if you are in the

list. (I had to remove the list because it attracts too many false search result clicks - like black SEO of sorts)

Query strings used (includes Partner / affiliate IDs - who gets paid for this traffic. The number in brackets shows the number of times it was used) (I had to remove the list because it attracts too many false search result clicks - like black SEO of sorts)