

The Ghost Dragon

Introduction

Cylance SPEAR™ has identified an APT group which deploys multiple customized malware implants, targeting mainly Chinese and Russian users. Cylance determined that the 'Ghost Dragon' group utilized specifically tailored variants of [Gh0st RAT](#), which the group modified from the 3.6 version of the source code released in 2008. Newly implemented security mechanisms in the altered malware makes identification of Gh0st RAT Command and Control network traffic more difficult for both security products and researchers.

This write-up provides initial disclosure of a portion of the malware and infrastructure used by the Ghost Dragon group and covers the new security mechanisms in detail, as well as revealing how researchers were able to communicate with the custom implant by rebuilding and compiling a customized Gh0st RAT controller.

The Standard Gh0st RAT Protocol

The standard network protocol for Gh0st RAT 3.6 employs zlib compression, which utilizes 'Gh0st' as a static five-byte packet flag that must be included in the first five bytes of initial transmission from the victim (as seen in Figure 1). During the initial login request, the 3.6 version of Gh0st RAT enumerates system information and transmits that information to the controller. The proceeding eight bytes of the packet contain both compressed and uncompressed size information followed by the zlib compressed data, starting with bytes 78 9C at offset 14 (shown in Figures 1 and 2, below).

```

47 68 30 73 74 a4 00 00 00 e0 00 00 00 78 9c 4b Gh0st... ..x.K
cTc.....@....
....S..2.S.....
...d.,. ,...y
k.....`...'
..4.... .p.+...
...'..80.....5:
.....5.. ..^[...
..... n `dd8Q..
.....T.R..
..).'

```

Figure 1: Standard Gh0st RAT 3.6 packet login

request | Victim to controller

After a successful login, the controller returns an acknowledgment to the victim with the correct header using the packet flag which it is programmed to verify, as seen in Figure 2. The connection will be established only if the packet flags match.

```

47 68 30 73 74 16 00 00 00 01 00 00 00 78 9c 63 Gh0st... ..x.c
00 00 00 01 00 01 .....
.....

```

Figure 2: Standard Gh0st RAT 3.6 packet response | Controller to victim – 22 bytes

Following the first five bytes, the packet then shows the full packet size and the uncompressed data size. The Gh0st packet header for version 3.6 uses the 13-byte format shown in Figure 3:

5 byte 'G' 'h' '0' 's' 't' | 4 byte PacketLen | 4 byte Unzip Len |

Figure 3: Gh0st RAT 3.6 header format

After the packet flag, packet length and unzip length have been verified, a working connection is created between the victim and the controller. Subsequent keep-alive transmissions ensure a continuous connection.

Once Gh0st RAT is connected, the attacker has full remote administration tool (RAT) functionality via the controller. Available remote functionality includes:

- • **Fully functional remote desktop display, including the ability to remotely block user input**
- • **Fully functional file manager that lists local and network drives**
- • **Ability to interact with running processes**
- • **Ability to interact via the command console**
- • **Key logging**
- • **Audio capture**
- • **Webcam capture**
- • **Ability to send 'Windows style' alerts to the victim, which can display any text the attacker enters into the controller**

These features ensure complete remote control of the computer.

Changes to the Gh0st RAT Protocol

In an older version of the customized Gh0st RAT malware, the protocol packet flag is no longer represented by the string 'Gh0st'. The Ghost Dragon group modified the source code and changed the packet flag to 'XYTvn', as seen in Figure 4. The packet structure still implements the same 13-byte header format, including the starting zlib compression bytes of '78 9C':

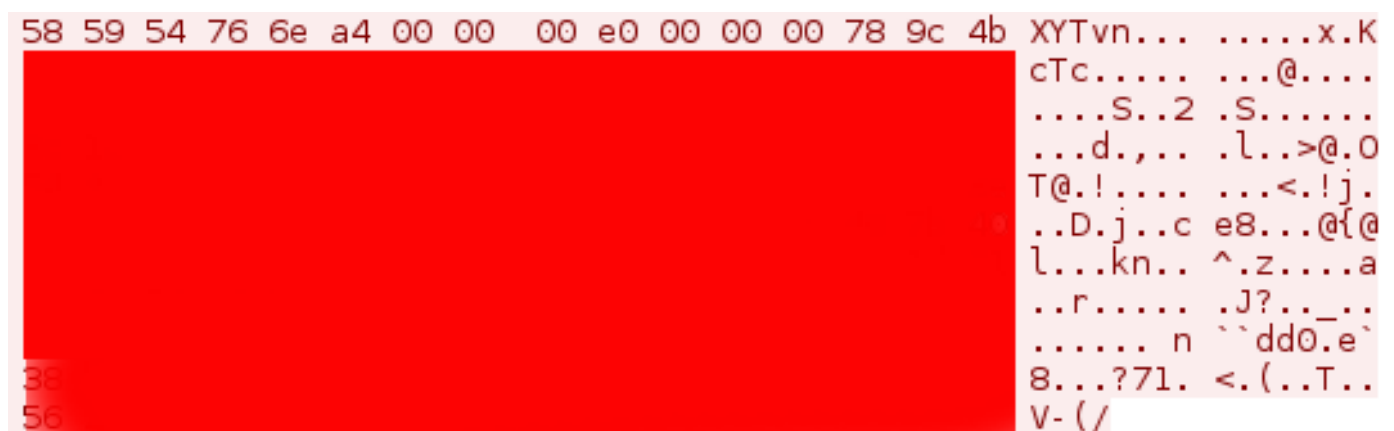


Figure 4: XYTvn static packet flag | Victim to controller login request

Sample: f9a669d22866cd041e2d520c5eb093188962bea8864fdfd0c0abb2b254e9f197

In a more recent version of the modified Gh0st RAT malware, Ghost Dragon implemented dynamic packet flags which change the first five bytes of the header in every login request with the controller. This complicates identifying its network traffic, as the header bytes in the zlib compressed data section no longer start with '78 9C', as shown in Figure 5.

In some cases this was achieved through a simple XOR obfuscation of the packet data, beginning with the zlib header. SPEAR has observed numerous different XOR keys utilized by the group. No changes to the compression have been found; however, it is trivial to implement a modified compression protocol.

```

69 71 71 44 69 b6 00 00 00 f8 00 00 00 4a ae 79 iqqDi... .....J.y
[REDACTED] QRR..WR. ...BR.w.
[REDACTED] !.-0V.2. 2...1...
[REDACTED] f.*9q... p?..:10.3
[REDACTED] .....".". ...6.S.(
[REDACTED] Z.:.R.U. V.7P.414
[REDACTED] !..3q?.U ..#...E.
[REDACTED] .....'. ...3.)..o
[REDACTED] .Z..... q6....5.
[REDACTED] >R...4.g af.c.lq.
[REDACTED] .5.r..b. .."....j
31 00 60 19 26 f3 1.`.&.

```

Figure 5: Dynamic packet flag | Victim to controller login request

Sample: 1be9c68b31247357328596a388010c9cffadcb6e9841fb22de8b0dc2d161c42

(Note: At the time of this report, the C2 for the sample was active and the malware could still establish an active connection to the Ghost Dragon controller at `bbs.winupdate[dot]net`. Currently, the domain `bbs.winupdate[dot]net` resolves to `122.10.18[dot]166`).

The reply from the active command and control server can be seen in Figure 6:

```

69 71 71 44 69 16 00 00 00 01 00 00 00 4a ae 51 iqqDi... .....J.Q
00 00 00 33 00 01 ...3..

```

Figure 6: Dynamic packet flag | Controller to victim login reply

Sample: 1be9c68b31247357328596a388010c9cffadcb6e9841fb22de8b0dc2d161c42

Connecting to the Ghost Dragon Malware

After successful identification of the malware as Gh0st RAT and confirmation of the modified command and control protocol, the final step toward verification was to connect with the malware to ensure that it would parse the normal Gh0st RAT commands.

I was able to compile a custom controller from the source code for the purposes of testing the Ghost Dragon malware. I bypassed the header checks looking for the 'Gh0st' packet flag, and reprogrammed the standard Gh0st RAT controller to reply with the packet flag sent from the victim.

I used sample **6c7f8ba75889e0021c4616fcbee86ac06cd7f5e1e355e0cbfbbb5110c08bb6df** for testing, which was executed by **f9a669d22866cd041e2d520c5eb093188962bea8864fdfd0c0abb2b254e9f197** during analysis on Windows XP. This sample transmitted the static five byte packet flag 'XYTvn'.

I ran the victim malware and changed the host's file to point the domain to a host-only IP address. I received a connection attempt from the victim and a response from the controller returned the correct packet flag. Afterward, the keep-alive packets between the victim and controller maintained the connection, which allowed me to test the functionality of **6c7f8ba75889e0021c4616fcbee86ac06cd7f5e1e355e0cbfbbb5110c08bb6df** with the custom Gh0st RAT controller.

The following screen shots were obtained while connected to the malware:

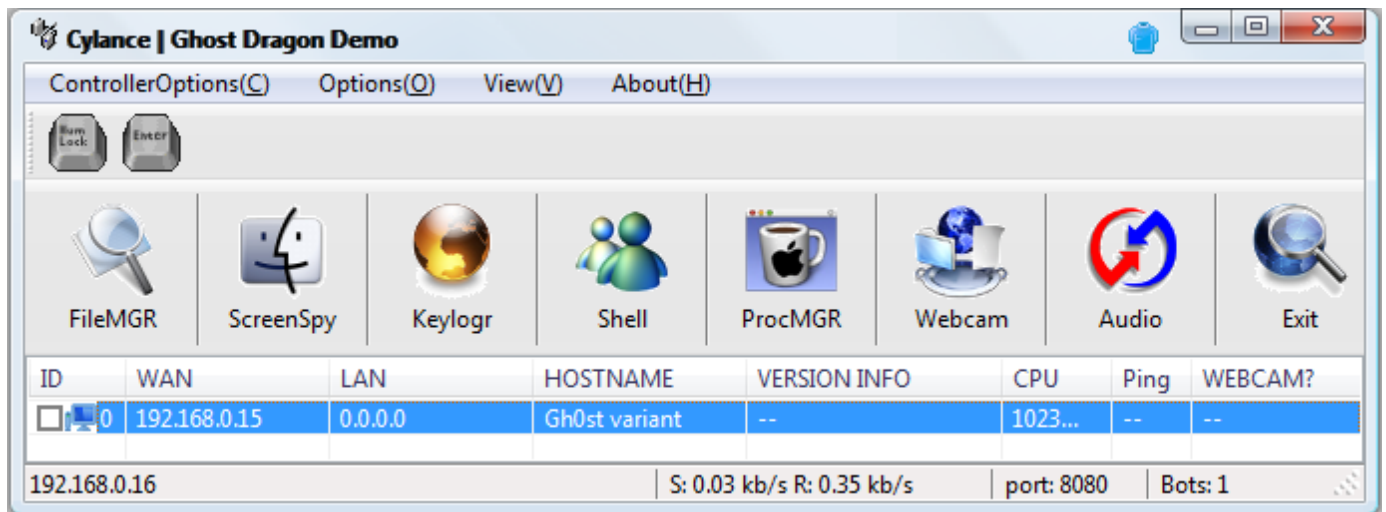


Figure 7: Main connection view | Connected to 6c7f8ba75889e0021c4616fcbee86ac06cd7f5e1e355e0cbfbbb5110c08bb6df

The HOSTNAME field shown in Figure 7 was programmed in my custom controller to display 'Gh0st variant', just in case the incoming structure parsed from the victim did not match the format of the default 'LOGININFO' structure from Gh0st RAT 3.6.

I attained full functionality using the Gh0st 3.6 protocol in the controller, despite the fact that the incoming structure was not exactly what my customized controller expected. As shown in Figure 8, I successfully added an administrative user while remotely connected via my custom Gh0st RAT console within the malware dropped from sample f9a669d22866cd041e2d520c5eb093188962bea8864fd0c0abb2b254e9f197.

Figure 8 demonstrates that some of the characters in the window are not in English and are not displayed properly on the form. This is due to the fact that the characters were written in Mandarin in the original Gh0st RAT 3.6 source. I changed phrases in the main connection window (Figure 7) for demonstration purposes only.

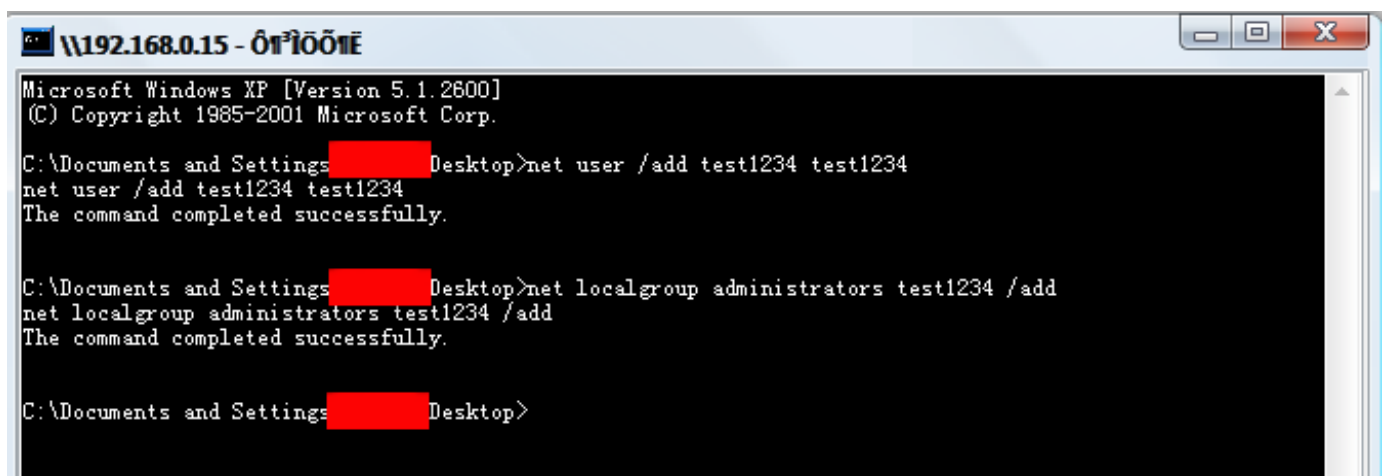


Figure 8: Remote shell | Adding admin user | Connected to 6c7f8ba75889e0021c4616fcbee86ac06cd7f5e1e355e0cbfbbb5110c08bb6df

The remaining features of the Gh0st RAT protocol were tested successfully with the custom Gh0st RAT controller while connected to

6c7f8ba75889e0021c4616fcbee86ac06cd7f5e1e355e0cbfbbb5110c08bb6df.

Additional analysis into other samples is ongoing and more information will be forthcoming.

Network IOCs and Infrastructure Overlap

A48f881f254dc8452561a8f13e2fb81933473ff22e549787f0ca67f19ba7fe67

File name: Air China 2015 April TIMETABLE .xls

Malware type: Initial Infection Vector/ XLS file

Network Activity Summary: Drops the downloader

71a52058f6b5cef66302c19169f67cf304507b4454cca83e2c36151da8da1d97

71a52058f6b5cef66302c19169f67cf304507b4454cca83e2c36151da8da1d97

File name: AdobeWpkReg.tmp

Malware type: Downloader

Network Activity Summary: Uses HEAD method (instead of GET) | Calls out to info.winupdate[dot]net/robots.txt | User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; NULL[5(1)]Windows NT 6.1; Trident/6.0)

Note: User Agent is variable depending on the version info of the host

1be9c68b31247357328596a388010c9cffadcb6e9841fb22de8b0dc2d161c42

File name: iconfig.exe

Malware type: Gh0st RAT variant

Network Activity Summary: bbs.winupdate[dot]net | Port 8080

Packet flag: Dynamic

f9a669d22866cd041e2d520c5eb093188962bea8864fdfd0c0abb2b254e9f197

File name: install.exe

Malware type: Gh0st RAT variant

Network Activity Summary: ooxxxoo.gicp[dot]net | Port 8080 | Also connects to www.winupdate[dot]net on port 8080

Packet flag: Static: XYTvn

Note: On Windows XP, may drop a replacement DLL for the AppMgmt service using ClimateVMMain export SVC_sha256: 6c7f8ba75889e0021c4616fcbec86ac06cd7f5e1e355e0cbfbbb5110c08bb6df. The hash of the dropped file may change on different versions of Microsoft Windows

99ee5b764a5db1cb6b8a4f62605b5536487d9c35a28a23de8f9174659f65bcb2

File name: install.exe

Malware type: Gh0st RAT variant

Network Activity Summary: www.searchhappynews[dot]com | Port 80

Packet flag: Static | XYTvn

b803381535ac24ce7c8fdcf6155566d208dfca63fd66ec71bbc6754233e251f5

File name: ExtensionManager.exe

Malware type: Gh0st RAT variant

Network Activity Summary: www.fhtd[dot]info | Port 1081

Packet flag: Dynamic

Infrastructure Overlap

Domain	 Previous IP resolution
bbs.winupdate[dot]net	122.10.18.166
www.fhtd[dot]info	122.10.18.166
info.winupdate[dot]net	 122.10.36.94

During investigation into the infrastructure used by Ghost Dragon, an anonymous FTP server was discovered on one of the IP addresses listed above, which hosted 'info.winupdate[dot]net | 122.10.36.94'.

Upon obtaining the files on the FTP server, an older Gh0st RAT variant was obtained with the file name 'operas.exe'. This variant sends the system info to the controller in clear text for the login request.

fb5a7cb34040b1e98b077edaf91cb59a446d8ff07263afe875cf6bd85bfb359d

File name: operas.exe

Malware type: Gh0st RAT variant

Network Activity Summary: www.swgabeg[dot]com | Port 1080 | Clear text login request sent to controller

Packet flag: Dynamic

Appendix A – IP and Domain Listing **IP addresses:**

101.55.33.39
103.232.215.144
103.246.245.147
111.68.8.130
112.125.17.103
113.10.148.161
113.10.148.205
122.10.18.166
122.10.36.94
122.10.41.85
122.10.83.75
122.10.85.35
122.9.247.128
122.9.247.134
122.9.247.216
122.9.247.56
123.254.111.87
142.4.103.90
174.128.255.228
175.45.192.234
202.172.32.172
202.174.130.116
203.232.28.10
209.85.84.165
209.85.84.167
31.170.179.179
58.64.187.22

60.215.128.246

64.111.220.218

Domains:

info.winupdate[dot]net

bbs.winupdate[dot]net

ooxxxoo.gicp[dot]net

[www.winupdate\[dot\]net](http://www.winupdate[dot]net)

[www.searchhappynews\[dot\]com](http://www.searchhappynews[dot]com)

[www.fhtd\[dot\]info](http://www.fhtd[dot]info)

www.swgabeg[dot]com

