BabyShark Malware Part Two – Attacks Continue Using KimJongRAT and PCRat

By Mark Lim

Published: 2019-04-26 · Archived: 2026-04-05 20:55:22 UTC

Executive Summary

In February 2019, Unit 42 published a blog about the BabyShark malware family and the associated spear phishing campaigns targeting U.S. national think tanks. Since that publication, malicious attacks leveraging BabyShark have continued through March and April 2019. The attackers expanded targeting to the cryptocurrency industry, showing that those behind these attacks also have interests in financial gain.

While tracking the latest activities of the threat group, Unit 42 researchers were able to collect both the BabyShark malware's server-side and client-side files, as well as two encoded secondary PE payload files that the malware installs on the victim hosts upon receiving an operator's command. By analyzing the files, we were able to further understand the overall multi-staging structure of the BabyShark malware and features, such as how it attempts to maintain operational security and supported remote administration commands. Based on our research, it appears the malware author calls the encoded secondary payload "Cowboy" regardless of what malware family is delivered.

Our research shows the most recent malicious activities involving BabyShark malware appear to be carried out for two purposes:

- Espionage on nuclear security and the Korean peninsula's national security issues
- Financial gain with focus on the cryptocurrency industry based on the decoy contents used in the samples, shown in Figure 1. Xcryptocrash is an online cryptocurrency gambling game.

Figure 1. Cryptocurrency related BabyShark malicious document decoy

We presume that the BabyShark malware toolset is shared among actors under the same umbrella or the same group has been assigned an additional mission.

In our analysis, we found BabyShark attacks were using KimJongRAT and PCRat as the encoded secondary payload and thus were the “Cowboys”.

Suspicious Access Logging

BabyShark has a multi-stage infection chain with checks between each stage, as shown in Figure 2, to ensure only targeted hosts are advanced to the next stage before it finally beacons backs to the attacker.

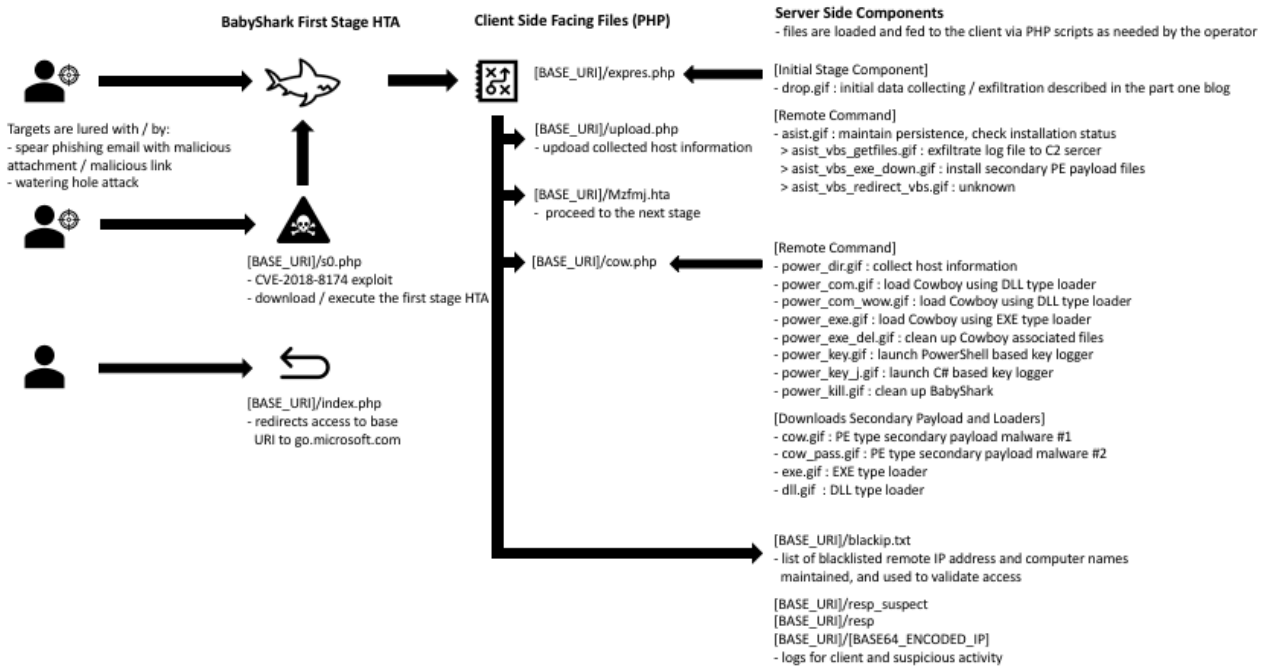

Figure 2. BabyShark malware overall structure

This is done by maintaining a list of denylisted IP addresses and computer names for those who have made suspicious access attempts, such as access with invalid parameters, to the server as a possible technique meant to make analysis harder. The IP addresses and computer names in the denylist are written in base64 encoded format at [BASE_URI]/blackip.txt, shown in Figure 3.

```
NT   4Tk5I     S4xMjA=
MT   LjIyN     C4xNTA=
NT   4Tk5I     S4xMjA=
MT   4jIzI     S4xMTU=
SQ   ADEAA     A==
MT   LjkzI     S4zOA==
Vw   AE4AA     DIAQgBJAFQALQBMAC0AMAA=
MT   Ljc2I     zI=
VQ   AEUAQ     FAAQwA=
MS   0TIuN     jE2OA==
Vw   AE4AI     EEASQAxAEQASQBRAEkANwBOAE4A
MT   LjE1I     y4xMDQ=
NT   4jA1I     jE4Ng==
Uw   AHMAc     G0ASQBUAA==
MT   LjIyN     DEuNjU=
Sg   AEUAQ     EcARQAtAFAAQwA=
NT   4Tc0I     y40Nw==
MT   LjQ3I     S4xOA==
Vw   AG4ALQBiAGUAcgBlAG4AYQA=
```

Figure 3. Denylisted IP addresses and computer names in blacktip.txt

When a new access attempt is made with data matching the denylist, the server will not proceed to the next stage and alerts the operator via a separate log file shown in Figure 4.

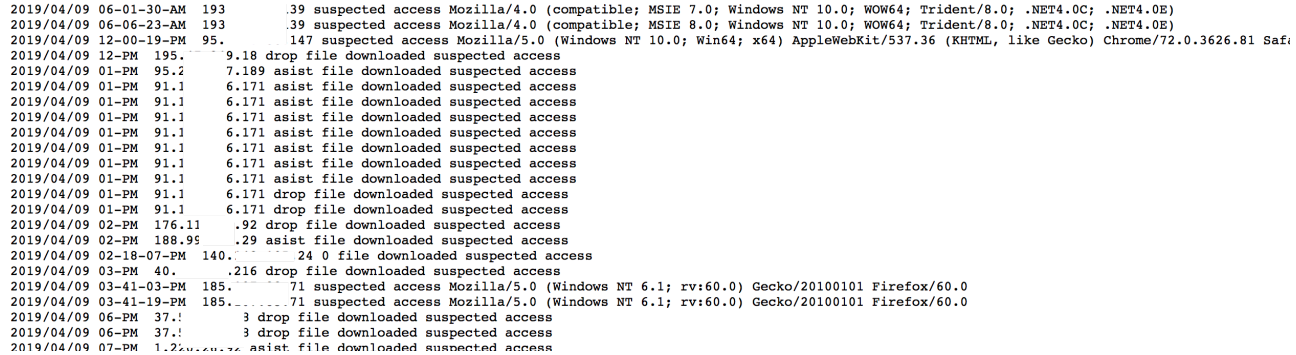

Figure 4. Suspicious activity log report to operator

BabyShark's C2 server also logs access to its base URI and redirects to http://go.microsoft[.]com/. The purpose of this is likely to avoid its files being seen due to potential mis-configurations of the hosting web server.

```
if($ff=fopen("resp_suspect","a"))

{

fwrite($ff, $date . "  " . $ip . " suspected access " . $useragent ."\r\n");

    fclose($ff);

}

header('Location: http://go.microsoft[.]com/');

exit;
```

Remote Commands

The operator can issue VBS and PowerShell based commands to victim systems infected with BabyShark. The remote commands we found from the C2 are in the below table, but BabyShark is not limited to these as the attacker can create more VBS or PowerShell command files.

VBS based remote commands:

Command Name	Description
getfiles	Archive all files in the BabyShark base path as a ZIP archive, then upload to the C2

exe_down	Download files for secondary payload: - a Cowboy, a custom encoded PE payload - an EXE type loader which decodes and loads Cowboy in memory - a DLL type loader which decodes and loads Cowboy in memory
redirect_vbs	Purpose of this command is not clear as key file is missing, but it is likely for changing C2 path

Table 1. VBS based remote commands for BabyShark

PowerShell based remote administration commands:

Command Name	**Description**
keyhook	Two types of key loggers implemented using PowerShell and C# - PowerShell based key logger which is openly available on GitHub. Result is saved in %APPDATA%\Microsoft\ttmp.log - C# based key logger saves result in %APPDATA%\Microsoft\ttmp.log
dir list	Collect host information and save the result in %APPDATA%\Microsoft\ttmp.log. The commands issued to collect host information include: - whoami - hostname - ipconfig - net user - arp -a - dir "%appdata%\Microsoft" - dir "%systemroot%\SysWOW64\WindowsPowerShell\" - vol c: d: e: f: g: h: i: j: k: l: m: n: o: p: q: r: s: t: u: v: w: x: y: z: - dir "%userprofile%\Downloads" - dir "%userprofile%\Documents"

	- dir "%userprofile%\AppData\Local\Google\Chrome\User Data\Default" - tasklist Also, a test result for UAC accessibility, and Microsoft Office security setting from registry key values
power com	Copy %APPDATA%\Microsoft\delemd.tmp0 to %APPDATA%\Microsoft\XXYYZZ.tmp, and load as DLL
exe del	Clean up all files associated with secondary payload execution. - %APPDATA%\Microsoft\desktop.r3u, encoded Cowboy payload - %APPDATA%\Microsoft\fstnur, file used to check for first time execution - %APPDATA%\Microsoft*.tmp
execute	Copy %APPDATA%\Microsoft\deleme.tmp0 to %APPDATA%\Microsoft\deleme.tmp, and execute it

Table 2. PowerShell based remote commands for BabyShark

KimJongRAT and PCRat are the Cowboys!

The secondary malware is delivered as a set:

- one EXE loader
- one DLL loader
- one encoded payload

The functionality of the EXE and DLL loaders is the same: the only difference is the file type. These loaders are later run upon receiving an execution command: “execute” to invoke the EXE type loader or “power com” to launch the DLL type loader. We theorize the reason for having two different type loaders is to have redundancy for loading the payload in case of anti-virus software’s disruption. Either loader will load the custom encoded secondary payload, the Cowboy, in memory, decode it, and execute it.

In our previous research, we wrote about possible links between BabyShark and the KimJongRAT malware family. We based these possible links on the similarity of malware behavior, similar interests in the targets, and a freshly compiled KimJongRAT malware sample being seen from the same threat actor. In our latest analysis, we collected two secondary payload files, cow_pass.gif and cow.gif, from BabyShark’s C2 server. After decoding, we found these samples were KimJongRAT and PCRat respectively. Their metadata are in Tables 3 and 4.

SHA256	f86d05c1d7853c06fc5561f8df19b53506b724a83bb29c69b39f004a0f7f82d8
timestamp	2010-07-14 08:47:40

size	124,928
Import hash	d742aa65c4880f85ae43feebb0781b67
C2	173.248.170[.]149:80

Table 3. Decoded PCRat payload metadata

SHA256	d50a0980da6297b8e4cec5db0a8773635cee74ac6f5c1ff18197dfba549f6712
timestamp	2018-12-25 11:11:47
size	787,968
Import hash	daab894b81cc375f0684ae66981b357d

Table 4. Decoded KimJongRAT payload metadata

PCRat is an infamous remote administration trojan with its source code openly available on the public internet. The malware is a variant of the Gh0st RAT malware family and it shares many similarities with Gh0st including its network beacon structure as shown in the Figure 5.

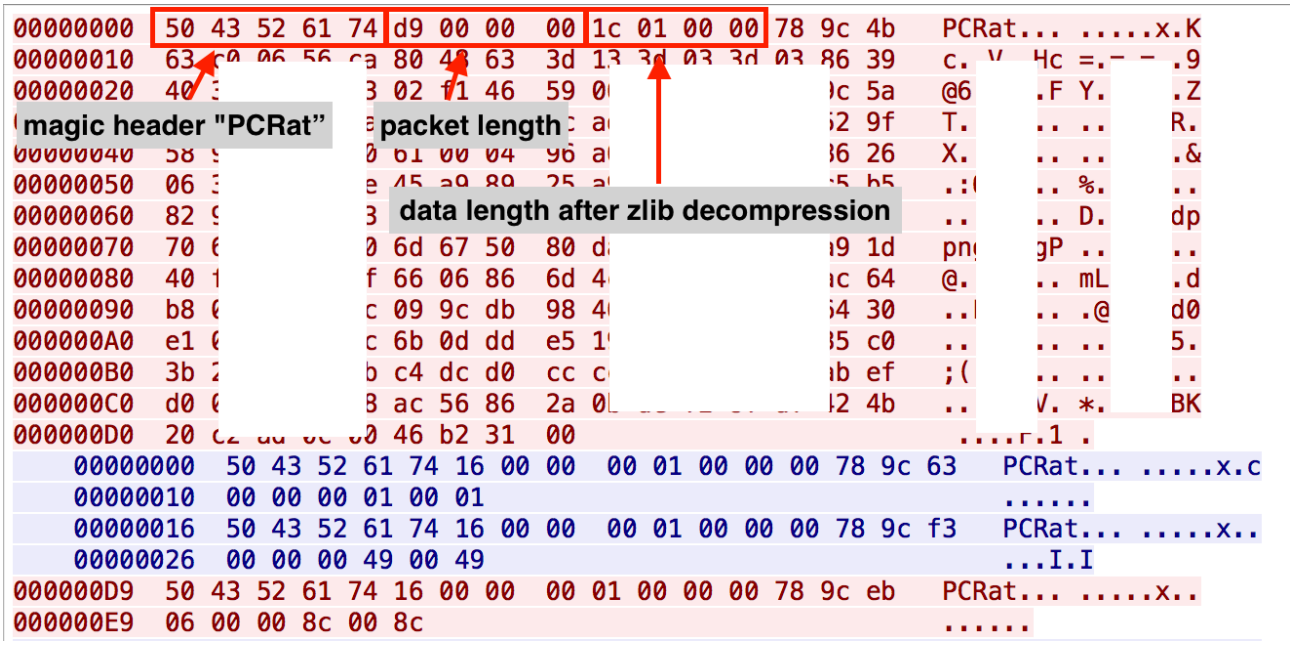

Figure 5. PCRat communication with the C2 at 173.248.170[.]149:80

Initially, we were curious about the sample's old timestamp and it being hardly modified from the original PCRat binary which had been publicly available for many years. However, the operator seemed to be actively operating the malware when we observed the communication between it and the C2 server at the time of our analysis.

The decoded KimJongRAT sample seems to exhibit a few changes in the code from the variants reported in the past. This sample added a substitution cipher to obfuscate API strings, as shown in Figure 5, to hide its intentions

and removed its networking feature for C2 data exfiltration, possibly in favor of the password gathering discussed below.

.rdata:0049E318	00000011	C	LxkArhygxDmzhgxV
.rdata:0049E32C	00000006	C	Jgxxi
.rdata:0049E334	0000000A	C	FriuCpgxV
.rdata:0049E340	00000014	C	VphxFdmoKrAygkpQukx
.rdata:0049E354	00000014	C	AygkpQukxKrVphxFdmo
.rdata:0049E368	00000013	C	LxkArhygxCpgxZmaxV
.rdata:0049E37C	0000000C	C	LxkCpgxJpnx
.rdata:0049E388	00000015	C	LxkVpzhrvjHpoxfkrouV
.rdata:0049E3A0	00000013	C	JxkCpgxMkkopqykxjV
.rdata:0049E3B4	00000009	C	OxmhCpgx
.rdata:0049E3C0	0000000C	C	FgrjxDmzhgx
.rdata:0049E3CC	00000011	C	KxoapzmkxIorfxjj
.rdata:0049E3E0	0000000D	C	FoxmkxKdoxmh
.rdata:0049E3F0	00000012	C	JxkKdoxmhIopropku
.rdata:0049E404	0000000F	C	FoxmkxIorfxjjV

Figure 6. Encrypted API strings in KimJongRAT

As the original filename “cow_pass.fig” suggests, KimJongRAT seems to be wholly used as a password extraction and information stealer tool by the threat actor, and the collected data are exfiltrated to C2 with support from other malware such as BabyShark or PCRat. The information that the KimJongRAT malware steals from victim machines include email credentials from Microsoft Outlook and Mozilla Thunderbird, login credentials for Google, Facebook, and Yahoo accounts from browsers Internet Explorer, Chrome, Mozilla Firefox, and Yandex Browser. All this information together with the victims machines' OS version are stored into the file "%APPDATA%\Microsoft\ttmp.log". The contents in "ttmp.log" always begin with the string "AAAAFFFF0000CCCC" and then appended with base64 encoded stolen credentials.

CVE-2018-8174

We have not observed an in-the-wild case yet, but we did find a PHP sample exploiting CVE-2018-8174 (Windows VBScript Engine Remote Code Execution Vulnerability) on the BabyShark C2 server, and this suggests that the threat actor may be leveraging this vulnerability to make a target load BabyShark’s first stage HTA via a watering hole attack or a malicious URL in a spearphishing email.

The attacker’s exploit script logs the victim’s remote IP address and redirects to http://google[.]com if the access is made more than one time from the same IP. This again is perhaps a tactic meant to thwart researchers.

```
if(file_exists($filename))

{

    if($ff=fopen("resp","a"))

    {
```

```
        fwrite($ff, $date . "  " . $ip .  "    ".$useragent."     reopen document." ."\r\n");
        fclose($ff);
    }
    header("location: http://google[.]com");
    exit;
}
if($ff=fopen("resp","a"))
{
    fwrite($ff, $date . "  " . $ip .  "    ".$useragent."           open document." ."\r\n");
    fclose($ff);
}
```

Cowboy Converter

During our research, we discovered a Graphical User Interface (GUI) based program likely created by the BabyShark malware author from a public malware repository. The file is to use as a file encoder tool to convert a PE file into a payload format loadable by the previously described Cowboy EXE and DLL loaders. We believe this tool is used by the BabyShark author to create their attack. Its metadata is in Table 5, below.

SHA256	bd6efb16527b025a5fd256bb357a91b4ff92aff599105252e50b87f1335db9e1
timestamp	2019-01-30 18:22:51
size	24,576
Import hash	bde663d08d4e2e17940d890ccf2e6e74

Table 5. Cowboy converter metadata

This tool simply opens a file with the name of “cowboy” in the current working directory and encodes it into the Cowboy encoding format as detailed below. If a file with the name of “cowboy” is not found, it pops up a message box notifying “The file cowboy isn’t there!” shown in Figure 7.

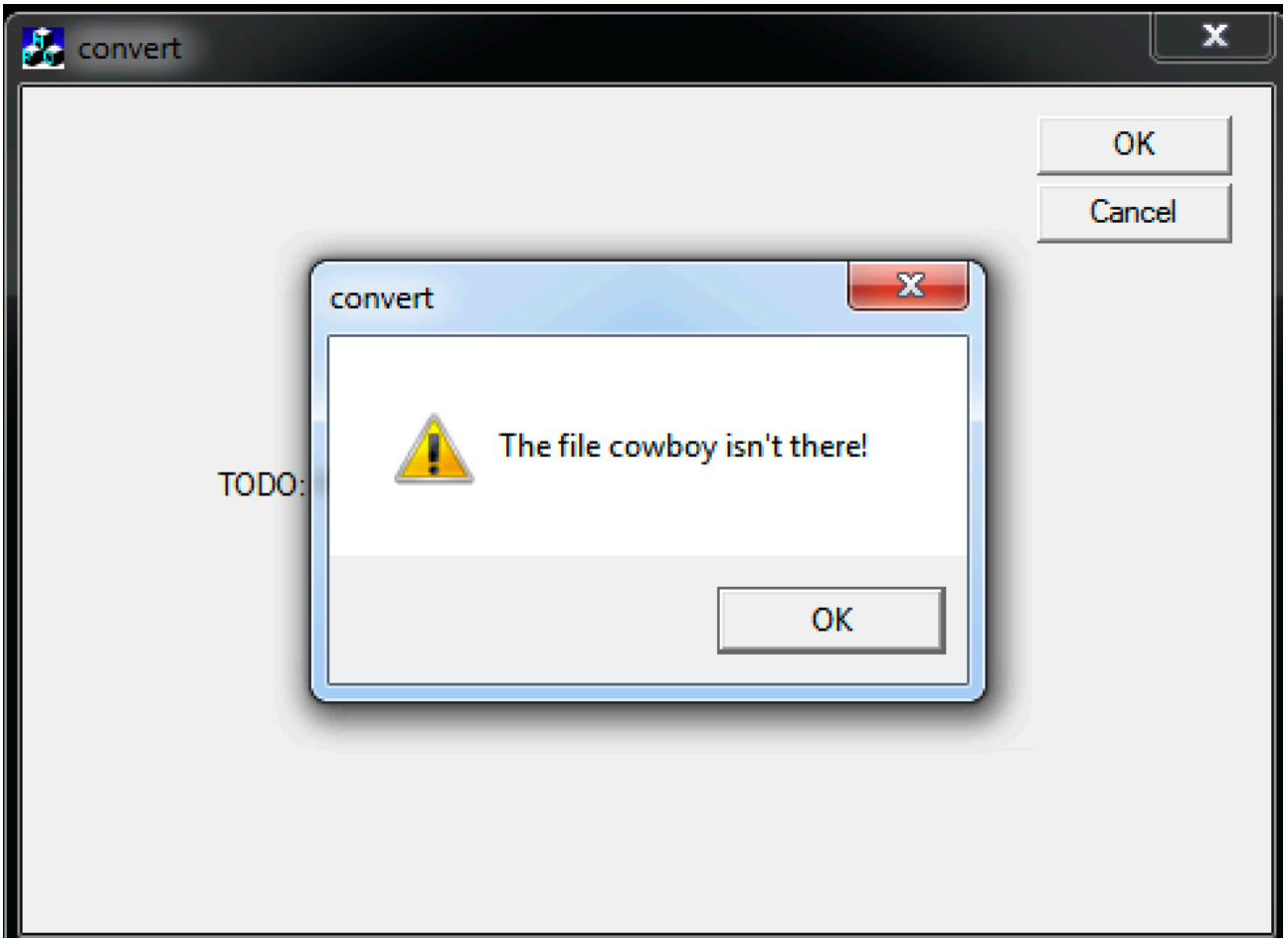

Figure 7. Cowboy converter and cowboy file not found pop up message

The encoding is done via the following three steps:

1. Reverse the original byte content read from the file with the name of “cowboy”
2. Take the reversed bytes and Base64 encode them
3. Take the base64 encoded string and chop it into 10 blocks and reverse the blocks' order

We have written a decoder script in Python and it is available in the appendix section of this blog.

Conclusion

Since releasing our previous research, malicious attacks leveraging the BabyShark malware have continued. In fact, they have widened their operation to target the cryptocurrency industry. The malware’s server-side implementation showed that the malware author has made certain efforts to maintain the operational security for operating the malware and C2 infrastructures. The threat actor leverages other commodity and custom developed tools in their campaigns. In this case, they were PCRat and KimJongRAT, but these may be changed to other malware families in the future. Malicious attacks using the BabyShark malware also seem likely to continue based on our observations and may continue expanding into new industries.

Palo Alto Networks customers are protected from this threat in the following ways:

- WildFire and Traps detect all malware families and vulnerability exploits mentioned in this report as malicious
- C2 domains used by the threat actors are blocked via Threat Prevention
- Pre and post infection network communications by the BabyShark and PCRat malware families are blocked by our IPS engine
- CVE-2018-8174 exploit is blocked by our IPS engine

AutoFocus customers can monitor ongoing activity from the threats discussed in this report by looking at the following tags:

- BabyShark
- CowboyLoader
- CowboyConverter

Palo Alto Networks has shared our findings, including file samples and indicators of compromise, in this report with our fellow Cyber Threat Alliance members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. For more information on the Cyber Threat Alliance, visit www.cyberthreatalliance.org.

Indicators of Compromise

Malicious Word Macro Document

75917cc1bd9ecd7ef57b7ef428107778b19f46e8c38c00f1c70efc118cb8aab5,

PCRat

f86d05c1d7853c06fc5561f8df19b53506b724a83bb29c69b39f004a0f7f82d8,

KimJongRAT

d50a0980da6297b8e4cec5db0a8773635cee74ac6f5c1ff18197dfba549f6712,

Cowboy Loader

4b3416fb6d1ed1f762772b4dd4f4f652e63ba41f7809b25c5fa0ee9010f7dae7

33ce9bcaeb0733a77ff0d85263ce03502ac20873bf58a118d1810861caced254

Cowboy Converter

bd6efb16527b025a5fd256bb357a91b4ff92aff599105252e50b87f1335db9e1,

Appendix - Python Script for Decoding Cowboy

```
import base64

with open('cowboy', 'r') as file_in, open('cowboy_clear.bin', 'wb') as file_out:
```

```
    EncStr = file_in.read()
    BlkSz = 10
    len_EncStr = len(EncStr)
    NonBlk10_ptr = len_EncStr - (BlkSz -1) * (len_EncStr // BlkSz)
    NonBlk10 = EncStr [:NonBlk10_ptr]
    result = ''
    EncStr = EncStr [NonBlk10_ptr::]
    #print EncStr
    x = range (-1,BlkSz-1)
    Blksize1 = len_EncStr // BlkSz
    for n in x:
        loop_buff1_ptr = n * (len_EncStr // BlkSz)
        loop_buff1 = EncStr [loop_buff1_ptr:loop_buff1_ptr+Blksize1]
        #print loop_buff1
        result = loop_buff1 + result
    result = result + NonBlk10
    clear = base64.b64decode(result)[::-1]
    print clear
file_out.write(clear)
```

Source: https://unit42.paloaltonetworks.com/babyshark-malware-part-two-attacks-continue-using-kimjongrat-and-pcrat/