NACHOCHEESE Malware Profile

Operational (OP) | Cyber Espionage (CE)

May 30, 2018 01:33:00 PM, 18-00006705, Version: 2 Risk Rating: HIGH

Executive Summary

- NACHOCHEESE is a TCP tunneler that is believed to have been part of the recent Society for Worldwide Interbank Financial Telecommunication (SWIFT) compromises throughout 2017 at Far Eastern International Bank (FEIB) in Taiwan and against Polish banks.
- It requires an IP address or domain to be passed to the program as an encoded parameter to be able to run.
- The code contains a series of unique Russian-language strings that are reportedly false-flags and has been observed being packed by Themida.

Analysis

CONTENTS

NACHOCHEESE Malware Summary

NACHOCHEESE is a command-line TCP tunneler implicated in bank heists from Lazarus group according to open-source reporting. Russian-language strings are hardcoded in NACHOCHEESE reportedly as false-flags. NACHOCHEESE accepts delimited command and control (C&C) communications in the form of IPs or domains as command-line arguments.

File Characteristics

File Name	File Type	Size	MD5	Compile Time
msmpeng.exe	PE.EXE (Packed)	1,637,888	3c9e71400b72cc0213c9c3e4ab4df9df	2017/02/20 11:09:30
splwow32.exe	PE.EXE	232,960	97aaf130cfa251e5207ea74b2558293d	2017/02/20 11:09:30
UNAVAILABLE	PE.EXE	522,160	284c1d29e54201447180dd174d9397e3	2016/08/26 04:11:49

UNAVAILABLE	PE.EXE	519,392	25200d3fe30785f3c90a91faf8ebf1b5	2016/08/26 04:11:49
UNAVAILABLE	PE.EXE	408,576	40e698f961eb796728a57ddf81f52b9a	2016/07/08 23:11:36
UNAVAILABLE	PE.EXE (Packed)	4,292,608	764d4c0b7abb95febd3cda88b46155dc	2017/02/20 11:09:30
UNAVAILABLE	PE.EXE	480,768	889e320cf66520485e1a0475107d7419	2016/08/26 04:11:49
UNAVAILABLE	PE.EXE	480,776	5994a8fd8c68dd1cc51ce7ca0d9c2749	2016/08/26 04:11:49

Table 1: Malware Characteristics

Persistence Mechanism

The malware does not contain a persistence mechanism. An external tool or installer is required if attackers desire persistence.

Host-Based Signatures

Unique Strings

kliyent2podklyuchit
ssylka
ustanavlivat
poluchit
pereslat
derzhat
vykhodit
Nachalo
Dazdrav$958478Zohsf9q@%5555ahshdnZXniohs
cEzQfoPw

NACHOCHEESE Technical Analysis

NACHOCHEESE is a TCP tunneling program that connects to the C&C server specified via a command-line parameter and relays its traffic to an endpoint that can be configured via a command-line parameter or via a command sent by the C&C server. The program accepts one or more pipe-delimited parameters via a single command-line parameter with the format specified in Figure 1. A sample command-line parameter is shown in Figure 2.

<C&C address>|<endpoint address>|<proxy address>|<proxy username:password>

Figure 1: Command line parameters

Encoded:

17210B264C09391D517440|02690B3B160F781304214A625A|1336172A1B44351F0676486A5252|1
6371D20581A370318

Decoded:

test.com:80|a-site.com:80|proxy.com:8080|user:pass

Figure 2: Sample command line parameter

Only the first parameter is required, the other three are optional. However, the parameters are position dependent. Addresses must be in the format <host:port>. Each parameter is encoded with an algorithm described in the Python snippet (Figure 3).

```
import binascii
import sys

c2 = bytearray(sys.argv[1])

key = bytearray("cEzQfoPw")
kLen = len(key)
k=0
out=""
for byte in c2:
    byte^=k
    byte= byte ^ (key[k % kLen])
    k+=1
    out+=chr(byte)

print "Commandline C&C: "+binascii.hexlify(out).upper()
```

Figure 3: Command-line parameter encoding function (Python)

Network-Based Signatures

The program initiates a handshake sequence whenever it requests a command from the client or requests traffic to be relayed:

1. The program attempts to connect to the C&C server. If a proxy address is specified, libcurl is used to attempt to connect (a sample of the HTTP request is shown in Figure 4. Figure 5 shows the HTTP request sent if a proxy username and password is configured).
2. Upon a successful connection, the program sends an encoded challenge request packet consisting of:
    a. 30 to 100 bytes of random data represented in Unicode hexadecimal characters
    b. Followed by an MD5 hash value of the Unicode string: Dazdrav$958478Zohsf9q@%5555ahshdnZXniohs\x00.
    c. Followed by the bytes 02 00
    d. This data is encoded in and encoded with a simple, custom XOR and shift-based encoding scheme that preserves the null byte values. A Python snippet to decode such a cipher text is shown in Figure 6. This equates to a payload size of 186 to 466 bytes: 46 to 116 bytes of data, multiplied by four for the Unicode hexadecimal encoding, plus two bytes for the trailing terminating bytes. A sample encoded payload of a challenge request packet is shown in Figure 7.
3. The program receives a challenge response from the C&C server, which is expected

to be encoded in the same way, with the same algorithm, and end with the same MD5 string and terminating bytes.

4. Once the response is verified, the length of the string kliyent2podklyuchit, which transliterates to "client to connect," is sent to the C&C server, followed by the string itself after being encoded using the same algorithm demonstrated in Figure 6.
5. Finally, the length of a message is sent, followed by the encoded message, to the C&C server. This message is Nachalo, which transliterates to "beginning," when the program is requesting a command from the C&C server, or ssylka, which transliterates to "link," when requesting traffic to relay to the endpoint.

```
CONNECT <addr:port> HTTP/1.1
Host: <addr:port>
```

Figure 4: Proxy CONNECT request

```
CONNECT <addr:port> HTTP/1.1
Host: <addr:port>
Proxy-Authorization: Basic <Base64 encoded username:password>
```

Figure 5: Proxy CONNECT request with authentication

```
import sys
from hashlib import md5

def decodeC2Buf(buf, key):
        out = ""
        for i in range(len(buf)):
                c = ord(buf[i])
                for j in range(len(key)-1, -1, -1):
                        c = ((c + ord(key[j])) & 0xFF) ^ ord(key[j])
                out += chr(c)
        return out

key = "47b0620e69f3228d6540bf3924a6c3bb8e68ebb5."decode("hex")
md5_salt = r"Dazdrav$958478Zohsf9q@%5555ahshdnZXniohs."encode("utf-
16le")+"\x00\x00"

with open(sys.argv[1], "rb") as rp:
        buf = rp.read()

dec_data = decodeC2Buf(buf, key)
rand_data, hash_val, trailer = dec_data[:-66], dec_data[-66:-2], dec_data[-2:]

print "Random Data:,"repr(rand_data)
print "Hash value:,"repr(hash_val)
print "Trailer:,"repr(trailer)
print "Valid Handshake:," (hash_val.decode("utf-16le") ==
md5(md5_salt).hexdigest().upper()
    and trailer == "\x02\x00")
```

Figure 6: C&C traffic decoding snippet (Python)

```
00000000  c5 00 f3 00 8e 00 7a 00  8b 00 8e 00 d9 00 fc 00   ......z. ........
00000010  ec 00 f3 00 ec 00 ec 00  09 00 b0 00 fe 00 6a 00   ........ ......j.
00000020  8b 00 fc 00 17 00 17 00  48 00 f3 00 8e 00 b0 00   ........ H.......
00000030  fe 00 a7 00 48 00 fe 00  fb 00 17 00 fc 00 c5 00   ....H... ........
00000040  b0 00 7a 00 6a 00 c5 00  d9 00 8b 00 6a 00 a7 00   ..z.j... ....j...
00000050  09 00 d9 00 a7 00 c5 00  09 00 48 00 48 00 09 00   ........ ..H.H...
00000060  f3 00 f3 00 c5 00 09 00  17 00 17 00 8b 00 a7 00   ........ ........
00000070  ec 00 48 00 c5 00 b0 00  d9 00 6a 00 7a 00 8b 00   ..H..... ..j.z...
00000080  f3 00 b0 00 17 00 7a 00  d9 00 fc 00 09 00 6a 00   ......z. ......j.
00000090  8b 00 a7 00 ec 00 09 00  fb 00 fc 00 6a 00 c5 00   ........ ....j...
000000A0  ec 00 f3 00 8e 00 b0 00  f3 00 f3 00 7a 00 48 00   ........ ....z.H.
000000B0  fb 00 7a 00 7a 00 c5 00  c5 00 f3 00 48 00 fc 00   ..z.z... ....H...
000000C0  6a 00 fc 00 d9 00 09 00  fc 00 c5 00 d9 00 fc 00   j....... ........
000000D0  f3 00 17 00 fc 00 48 00  fb 00 f3 00 ec 00 6a 00   ......H. ......j.
000000E0  09 00 fb 00 7a 00 8b 00  a7 00 a7 00 fb 00 b0 00   ....z... ........
000000F0  8e 00 c5 00 b0 00 48 00  17 00 c5 00 8b 00 6a 00   ......H. ......j.
00000100  8e 00 ec 00 f3 00 fe 00  d9 00 f3 00 a7 00 6a 00   ........ ......j.
00000110  ec 00 a7 00 b0 00 17 00  fc 00 48 00 48 00 09 00   ........ ..H.H...
00000120  09 00 09 00 48 00 8e 00  ce 00                     ....H... ..
```

Figure 7: Hex dump of sample encoded challenge request packet payload

Responses from the C&C server to the command request message (Nachalo) consist of a command keyword. Table 2 describes the supported commands.

Command	Translation	Description
ustanavlivat	To set	Set the endpoint to relay to
poluchit	To receive	Get the endpoint to relay to
pereslat	To send	Relay x number of messages (up to 0x2000 bytes each)
derzhat	To maintain	Do nothing, continue polling
vykhodit	To exit	Exit process

First Version Publish Date

April 25, 2018 11:14:00 AM

Tags HIGH

Threat Intelligence Tags

Malware Family

- NACHOCHEESE

Technical Indicators & Warnings

SHA1:	041c2a1a5810aa91788fc8b1604078c8133de1c3
Fuzzy Hash:	6144:sdqAqUok+00rm9TOi9Vc7/VtXvWLnJlh+efvoRKmjbL/xY4fTKKWSFle3IDgDi2j:xABogwttXuLnJlkkiKU/xtKYydF9iIUa
Identifier:	Attacker
File Name:	UNAVAILABLE
Malware Family:	NACHOCHEESE
File Size:	522160
File Compilation Date Time:	August 26, 2016 03:11:49 AM
SHA256:	9c62a70397370d5de689c66c667266ed4ebaa899d159a472c977fd94efcc1ef3
Type:	PE32+ executable (DLL) (GUI) x86-64, for MS Windows
MD5:	284c1d29e54201447180dd174d9397e3

SHA1:	723b77e6ae105d37e1f796646927748994f4b4dc
Fuzzy Hash:	49152:EbYVoaF2qjtw+sgEX0MrC2eXMUvrrs8OZxZI+wvTTahqO:PVDRtwTXvrC2eXdDAHZc
Packer:	Microsoft Visual C++ vx.x DLL,
File Name:	UNAVAILABLE
Malware Family:	NACHOCHEESE
SHA256:	91f72ce3dc835e372bbc73d5eb230604afc328519ed27a84ed105156ac1ea528
File Size:	4292608
File Compilation Date Time:	February 20, 2017 10:09:30 AM
Identifier:	Attacker
Type:	PE32 executable (GUI) Intel 80386, for MS Windows
MD5:	764d4c0b7abb95febd3cda88b46155dc

SHA1:	e0d5117ed5035658e9d5e67bc4b695958260eeaf
Fuzzy Hash:	6144:sdqAqUok+00rm9TOi9Vc7/VtXvWLnJlh+efvoRKmjbL/xY4fTKKWSFle3IDgDi2C:xABogwttXuLnJlkkiKU/xtKYydF9iIU
Identifier:	Attacker
File Name:	UNAVAILABLE
Malware Family:	NACHOCHEESE
File Size:	480776
File Compilation Date Time:	August 26, 2016 03:11:49 AM
SHA256:	aff19e65f87b672f7e89a4f07ae551a679bd2ceb2bb04814a79983ce77655457
Type:	PE32+ executable (DLL) (GUI) x86-64, for MS Windows
MD5:	5994a8fd8c68dd1cc51ce7ca0d9c2749

SHA1:	50b4f9a8fa6803f0aabb6fd9374244af40c2ba4c
Fuzzy Hash:	12288:E30MB7N+man4IrT0qhPyRg8o//ND6lAMYqcl:i0YNwrT0qhPFtHN2lLYq
Packer:	Microsoft Visual C++ 8,VC8 -> Microsoft Corporation,Microsoft Visual C++ 8,
File Name:	UNAVAILABLE
Malware Family:	NACHOCHEESE
SHA256:	a917c1cc198cf36c0f2f6c24652e5c2e94e28d963b128d54f00144d216b2d118
File Size:	408576
File Compilation Date Time:	July 08, 2016 10:11:36 PM

Identifier:	Attacker
Type:	PE32 executable (GUI) Intel 80386, for MS Windows
MD5:	40e698f961eb796728a57ddf81f52b9a
SHA1:	bdb632b27ddb200693c1b0b80819a7463d4e7a98
Fuzzy Hash:	24576:5gDgaE2r55ENJSOZ8jsAMZMF2kPupVevS6ieT17cZ/hJMIYO0:+D9vrrs8OZxZI+wvTTahqO
Identifier:	Attacker
File Name:	msmpeng.exe
Malware Family:	NACHOCHEESE
File Size:	1637888
File Compilation Date Time:	February 20, 2017 10:09:30 AM
SHA256:	70b494b0a8fdf054926829dcb3235fc7bd0346b6a19faf2a57891c71043b3b38
Type:	PE32 executable (GUI) Intel 80386, for MS Windows
MD5:	3c9e71400b72cc0213c9c3e4ab4df9df
SHA1:	1592ba6fcacee9e12ca09373f2264838139dafbb
Fuzzy Hash:	6144:sdqAqUok+00rm9TOi9Vc7/VtXvWLnJlh+efvoRKmjbL/xY4fTKKWSFle3IDgDi2C:xABogwttXuLnJlkkiKU/xtKYydF9iIU
Identifier:	Attacker
File Name:	UNAVAILABLE
Malware Family:	NACHOCHEESE
File Size:	519392
File Compilation Date Time:	August 26, 2016 03:11:49 AM
SHA256:	7d0ffd9fa1b21581fe9a1e1d852e348c83d4c3bbbd6751eb7f232e1bb27cc1ac
Type:	PE32+ executable (DLL) (GUI) x86-64, for MS Windows
MD5:	25200d3fe30785f3c90a91faf8ebf1b5
SHA1:	c7e7dd96fefca77bb1097aeeefef126d597126bd
Fuzzy Hash:	3072:6U5r72JE+FYWR0jZLShk4cPT/QzSaQ0sCFneZTznIhZJJcrJ1GHeV9:6U5uJpYnZL05STQNddFnAnGZIrV
Packer:	Microsoft Visual C++ 8,VC8 -> Microsoft Corporation,Microsoft Visual C++ 8,
File Name:	splwow32.exe
Malware Family:	NACHOCHEESE
SHA256:	9a776b895e93926e2a758c09e341accb9333edc1243d216a5e53f47c6043c852
File Size:	232960
File Compilation Date Time:	February 20, 2017 10:09:30 AM
Identifier:	Attacker
Type:	PE32 executable (GUI) Intel 80386, for MS Windows
MD5:	97aaf130cfa251e5207ea74b2558293d
SHA1:	f5fc9d893ae99f97e43adcef49801782daced2d7
Fuzzy Hash:	6144:sdqAqUok+00rm9TOi9Vc7/VtXvWLnJlh+efvoRKmjbL/xY4fTKKWSFle3IDgDi2C:xABogwttXuLnJlkkiKU/xtKYydF9iIU
Identifier:	Attacker
File Name:	UNAVAILABLE
Malware Family:	NACHOCHEESE
File Size:	480768

File Compilation Date Time:	August 26, 2016 03:11:49 AM
SHA256:	8cad61422d032119219f465331308c5a61e21c9a3a431b88e1f8b25129b7e2a1
Type:	PE32+ executable (DLL) (GUI) x86-64, for MS Windows
MD5:	889e320cf66520485e1a0475107d7419

Version Information

Version:1.0, April 25, 2018 11:14:00 AM
NACHOCHEESE Malware Profile

Version:2.0, May 30, 2018 01:33:00 PM
NACHOCHEESE Malware Profile

5950 Berkshire Lane, Suite 1600 Dallas, TX

75225

For more information please visit: https://intelligence.fireeye.com/reports/18-00006705