

# Parallax: The New RAT on the Block

By Arnold Osipov

Archived: 2026-04-05 18:19:30 UTC

Following the [increase](#) in **Parallax RAT** campaigns — *the new RAT on the block*, Morphisec Labs decided to release more technical details on some of the latest campaigns that the [Morphisec Preemptive Cyber Defense Platform](#) intercepted and prevented on our customer’s sites.

Parallax is an advanced remote access trojan that supports all Windows OS versions. It is capable of bypassing advanced detection solutions, stealing credentials, executing remote commands, and has also been linked to several [coronavirus malware campaigns](#).

Parallax is mostly delivered through malicious spam campaigns with Microsoft Word documents as the delivery vehicle of choice as will also be described in the following blog post.

## Technical Details

Before we dive into the details, we would like to cover the general flow of one of the attack chains we investigated.

### General Flow:

The first stage in this campaign is a Microsoft Word document with embedded macros. When macros are enabled, a DLL is dropped to the %Temp% directory. The export function of this dll is then invoked, which injects shellcode to the “*Notepad.exe*” process. This process is responsible for downloading the next stage from pastebin, which is the Parallax RAT loader.

The Parallax RAT loader does similar things in order to execute the final Parallax RAT payload. It injects a shellcode to the “*mstsc.exe*” process, which is responsible for downloading the next stage from “*i.imgur.com*” in the form of a picture. It then decrypts the picture and injects it into the “*cmd.exe*” process. As part of its persistence mechanism, scheduled tasks will be created to launch the malware at various intervals.

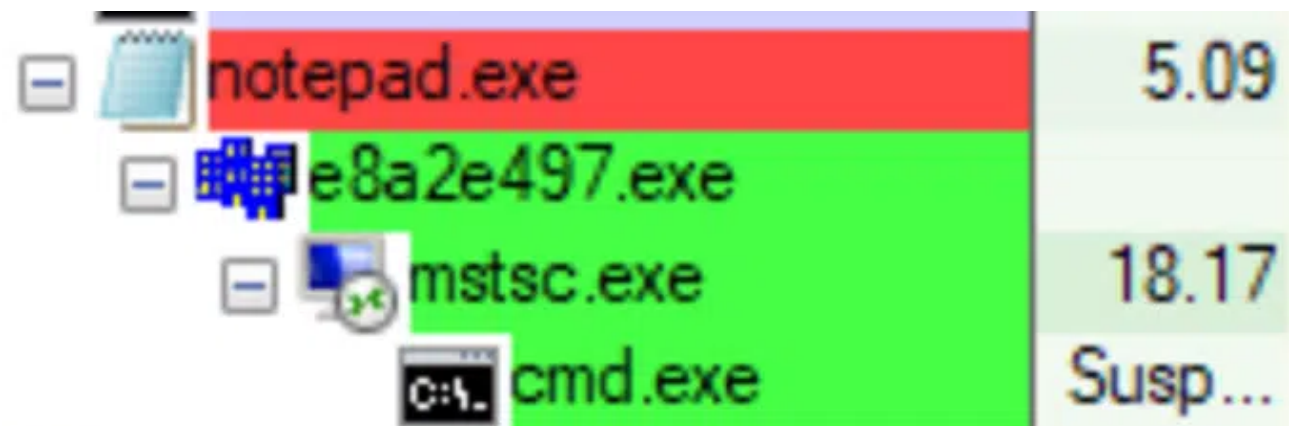


Figure 1 — The infection process tree

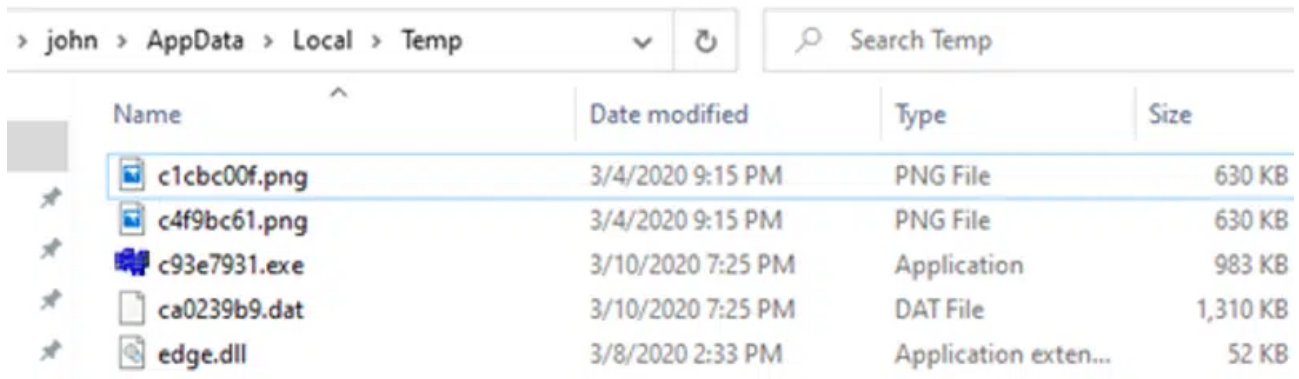


Figure 2 — The Parallax working directory

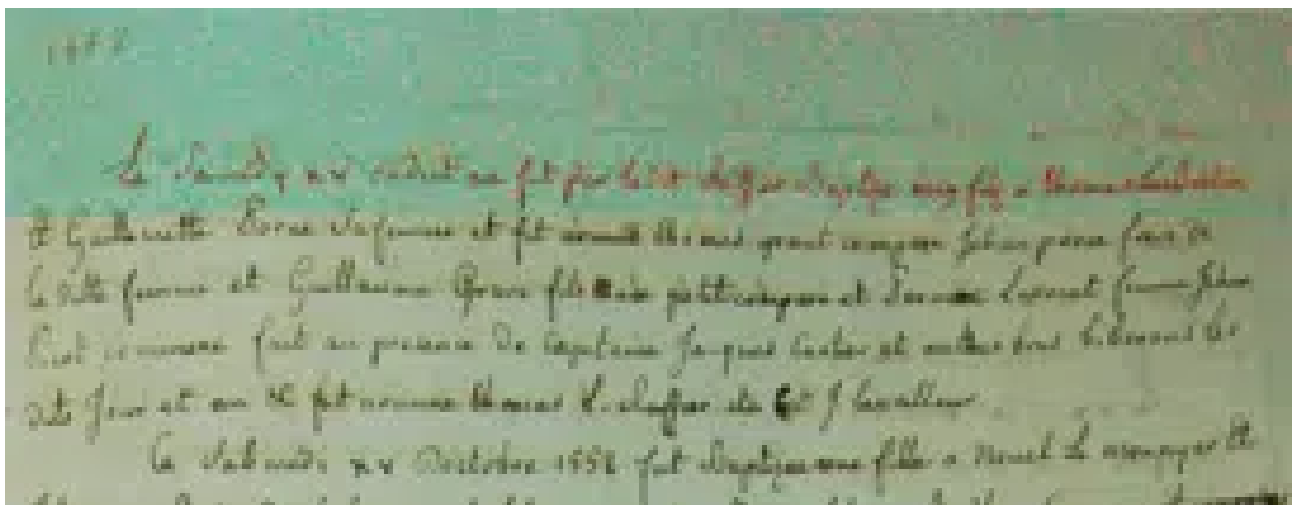


Figure 3 — A downloaded image from Imgur

## First Stage:

### Document:

Below is one example of a Microsoft Word document that's used to deliver Parallax RAT. Note the low detection rate according to VirusTotal. Morphisec Labs has seen these documents delivered via phishing emails to targeted machines since January 2020.



Figure 4 — The low detection rate in Virus Total

The content of the document is designed to lure the victim into enabling macros. Once that's done, the RAT can run and deliver its payload.



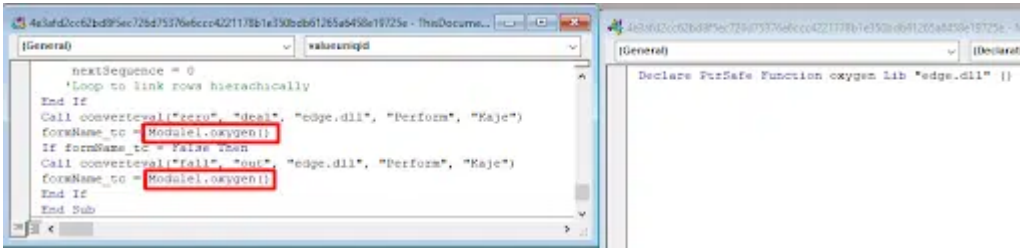


Figure 7 — The export function is invoked

The DLL export function is invoked after parsing completes.

**Second Stage:**

**DLL:**

The invoked DLL export function is responsible for decoding a shellcode that injects the next stage shellcode into a Notepad.exe process.

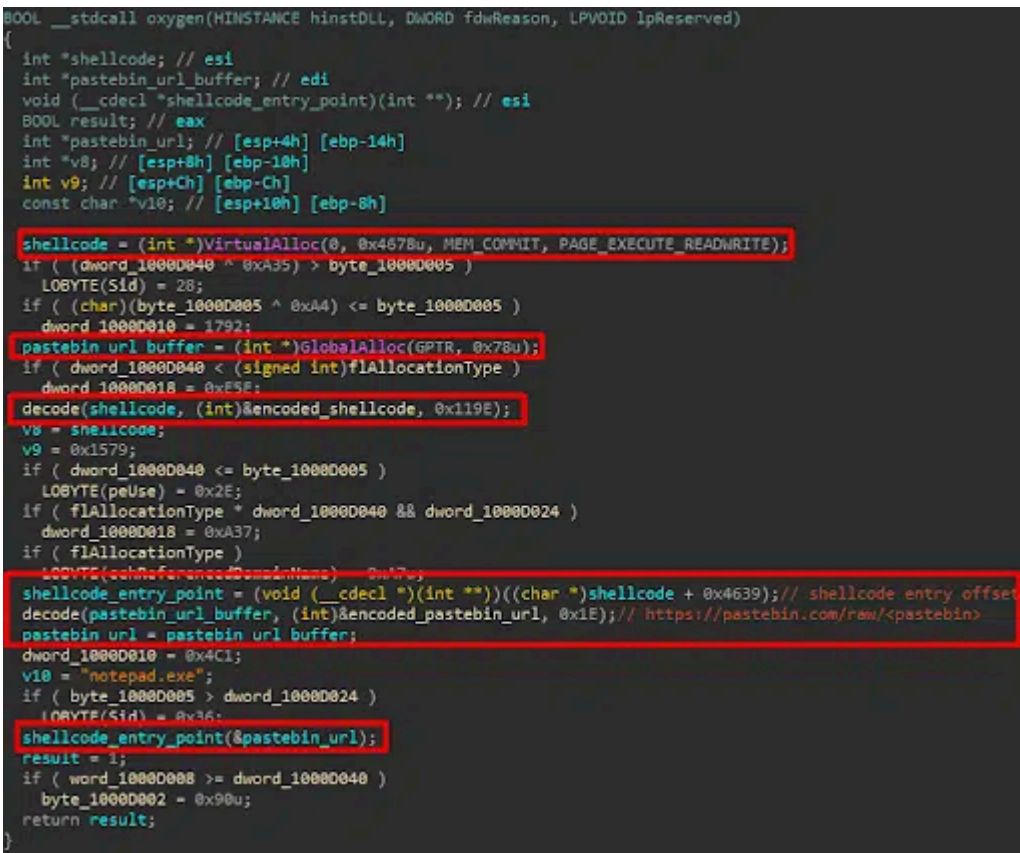


Figure 8 — The invoked DLL export function

invoked DLL export function

**First Stage Shellcode**

In order to hide the use of the low level (Nt\* and Zw\* functions) process hollowing injection, the shellcode uses direct syscalls. Attackers use this technique to escape debugger breakpoints as well as evade userland hooks. Parallax maps its own copy of ntdll into memory to utilize this technique.

```

1 int __cdecl map_ntdll(int ntdll_path, int hkernel32)
2 {
3     int (__stdcall *CreateFile)(int, MACRO_GENERIC, MACRO_FILE, _DWORD, signed int, MACRO_FILE, _DWORD); // 5728_4
4     int ntdll_filemap_handle; // eax
5     int (__stdcall *MapViewOfFile)(int, signed int, _DWORD, _DWORD, _DWORD); // [esp+8h] [ebp-18h]
6     int (__stdcall *CreateFileMapping)(int, _DWORD, signed int, _DWORD, _DWORD, _DWORD); // [esp+8h] [ebp-10h]
7     int ntdll_handle; // [esp+14h] [ebp-4h]
8
9     CreateFile = (int (__stdcall *))(int, MACRO_GENERIC, MACRO_FILE, _DWORD, signed int, MACRO_FILE, _DWORD)GetProcAddress_custom(hkernel32, 0xA1E9E29);
10    CreateFileMapping = (int (__stdcall *))(int, _DWORD, signed int, _DWORD, _DWORD, _DWORD)GetProcAddress_custom(
11        hkernel32,
12        0x48C273D);
13    MapViewOfFile = (int (__stdcall *))(int, signed int, _DWORD, _DWORD, _DWORD)GetProcAddress_custom(
14        hkernel32,
15        0xA29E382F);
16    ntdll_handle = CreateFile(ntdll_path, GENERIC_READ, FILE_SHARE_READ, 0, 0, FILE_ATTRIBUTE_NORMAL, 0);
17    if ( ntdll_handle == -1 )
18        return 0;
19    ntdll_filemap_handle = CreateFileMapping(ntdll_handle, 0, 0x1000000, 0, 0, 0);
20    return MapViewOfFile(ntdll_filemap_handle, 4, 0, 0, 0);
21 }

```

Figure 9 — Parallax maps its own ntdll copy to memory.

After the new copy of ntdll is mapped, Parallax uses simple offset extraction from the opcode to extract the system calls.

```

NtAllocateVirtualMemory = (int (__stdcall *))(signed int, _DWORD **)GetProcAddress_custom(hntdll, -668949132);
NtAllocateVirtualMemory_1 = from_ntdll_to_new_loaded_ntdll((int)NtAllocateVirtualMemory, hntdll, hntdll_mapped);
NtAllocateVirtualMemory_syscall = *((_DWORD *))(NtAllocateVirtualMemory_1 + 1); // offset to syscall (eax)

```

Figure 10 — syscall

extractions from suspicious functions.

```

> 009D1766 FF75 BC push dword ptr ss:[ebp-44]
  009D1769 49 dec ecx
  009D176A 5A pop edx
  009D176B 8B45 08 mov eax,dword ptr ss:[ebp+8]
  009D176E 83EC 28 sub esp,28
  009D1771 0F05 syscall

```

Figure 11 — Direct syscall invocation.

### Injected Shellcode

The injected shellcode (usually injected to Notepad.exe) is responsible for downloading and decoding Parallax RAT from pastebin.

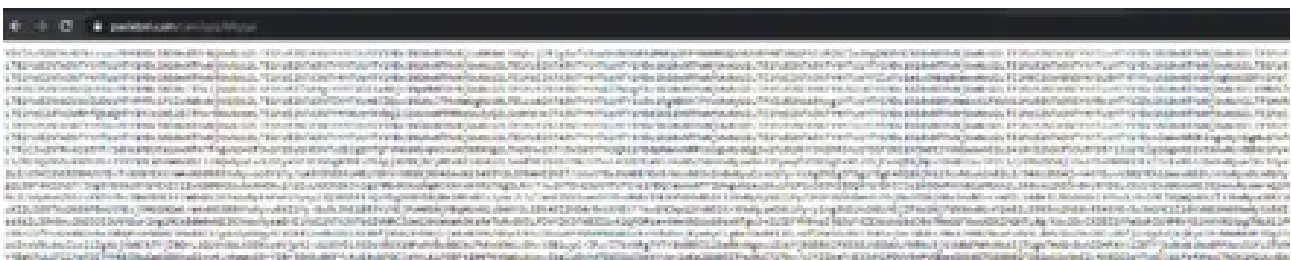


Figure 12 — Pastebin raw content

The pastebin content is decoded using base64 and XORed with a key that is generated using CRC32 checksum function on the pastebin URL.

```
1 int __cdecl decrypt(int encoded_content, int content_len, int key)
2 {
3     int result; // eax
4     int key_len; // [esp+0h] [ebp-8h]
5     int i; // [esp+4h] [ebp-4h]
6
7     result = strlenW(key);
8     key_len = result;
9     for ( i = 0; i < content_len; ++i )
10    {
11        result = *(char *)(key + i % key_len) ^ *(char *)(i + encoded_content);
12        *(_BYTE *)(i + encoded_content) = result;
13    }
14    return result;
15 }
```

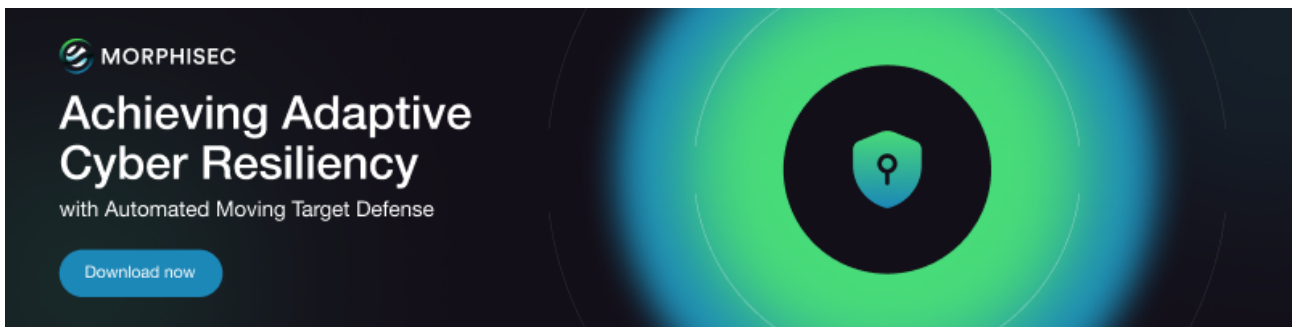
Figure 13 — The

Decoding routine.

The decoded Parallax payload is then dumped and executed from the %temp% directory. Vitali K covered the loader and image decoder that makes up Parallax RAT, while the pastebin decoder is accessible via Github at <https://gist.github.com/osipovar/a80e8b6b3caad209f17616761530302b>

## Conclusion

This new Parallax RAT campaign is indicative of the trend toward Malware-as-a-Service, or MaaS, one of the most pernicious weapons in the arsenal of threat actors. It’s also the trend that has largely driven the level of innovation in malware available to cybercriminals. Despite this, Morphisec customers can remain confident that they are protected against Parallax RAT and other remote access trojans through the power of [Automated Moving Target Defense](#).



## Appendix

IOCs:

Doc (SHA1):

- 2b2eaf94189d21b7a4418ff480fa332832aa0d98
- e793d2e0ac963357dc7895f62071c1036eba8284
- e440f67ca7d34be0f7346013d078072f64774e8c
- 45df85b3fe8954099cd49fdc5d59863baf1e6b76
- 40efa7e40846c5041e33ecd3396082a160f8d72c
- b4d8a4470ed1dc1dec7cf62c6d0bada7ca1fed21

- 242c71fda9c05f89730204361ff6a21cdae025e7
- 2ab5bae45055e0c18ac9f0ccc190f6f277dc806f
- ff8c49fbfb3da3a8e84bc332e646e4df3f3f6760
- 50c623fab59258300680f3dd0447cf3815498d89
- ff8c49fbfb3da3a8e84bc332e646e4df3f3f6760
- 161820606da9b7949dd45b93fe39b07b01bd973e
- 2fb1a63a3505427e42323bafef10349cc48b2a8b
- 420d9ffc0a760c40ca2e8ea480b8e268225a07f2
- 1dc94d5d49cd4ab215f291d188544a4996c05654
- 8642f6bb8b1db4c3adaad1c90167430f28536362

Pastebin:

- hxxps://pastebin[.]com/raw/2spx5VGG
- hxxps://pastebin[.]com/raw/5PiLyRjs
- hxxps://pastebin[.]com/raw/5UNceFha
- hxxps://pastebin[.]com/raw/aKj2aqwc
- hxxps://pastebin[.]com/raw/AvEEMK9J
- hxxps://pastebin[.]com/raw/BTiRSV6C
- hxxps://pastebin[.]com/raw/BXBbPstB
- hxxps://pastebin[.]com/raw/cpfstw2k
- hxxps://pastebin[.]com/raw/CuUTrPX0
- hxxps://pastebin[.]com/raw/drvV1FPJ
- hxxps://pastebin[.]com/raw/EnTPcdwc
- hxxps://pastebin[.]com/raw/exs0tSC7
- hxxps://pastebin[.]com/raw/FAUCzPvi
- hxxps://pastebin[.]com/raw/eYRSb32g

PNG:

- hxxps://i.imgur[.]com/02OZh3h.png
- hxxps://i.imgur[.]com/KPolbR1.png
- hxxps://i.imgur[.]com/s9Nu51u.png
- hxxps://i.imgur[.]com/bnFTfnL.png
- hxxps://i.imgur[.]com/B4MGZog.png
- hxxps://i.imgur[.]com/swnDCdS.png
- hxxps://i.imgur[.]com/H0RNHhb.png
- hxxps://i.imgur[.]com/4lQl9FZ.png
- hxxps://i.imgur[.]com/8kZ4rhJ.png
- hxxps://i.imgur[.]com/FGfZfCf.png
- hxxps://i.imgur[.]com/82WDYmV.png
- hxxps://i.imgur[.]com/aNQRmu1.png

**About the author**



Arnold Osipov

Malware Researcher

Arnold Osipov is a Malware Researcher at Morphisec, who has spoken at BlackHat and and been recognized by Microsoft Security for his contributions to malware research related to Microsoft Office. Prior to his arrival at Morphisec 6 years ago, Arnold was a Malware Analyst at Check Point.

---

Source: <https://blog.morphisec.com/parallax-rat-active-status>