Emotet Wi-Fi Spreader Upgraded

By Binary Defense

Archived: 2026-04-05 19:15:30 UTC

This an update to an early article regarding the emerging cyberthreat of Emotet Wifi Spreader.

Executive Summary

Binary Defense analysts previously discovered a stand-alone program for spreading Emotet infections over Wi-Fi networks. Although the spreader had been recently delivered by Emotet command and control (C2) servers, the program itself had not been changed for at least two years. In the last week, an updated version of the Wi-Fi spreader was observed being delivered to multiple bots. The new version changed the spreader from a stand-alone program into a full-fledged module of Emotet with some other functionality improvements. Instead of bundling the Emotet loader with the spreader, it now downloads the loader from a server.

Protocol Changes

While the changes to the Wi-Fi spreader do not affect the key functionality of the malware, the changes are still notable as they increase the logging capability of the spreader, allowing Emotet's authors to get step-by-step debugging logs from infected machines through the use of a new communication protocol.

This communication protocol uses two PHP POST arguments to provide Emotet's authors with crucial debugging outputs. The first argument, "id", is set to the victim's MachineGUID, while the second argument, "data" is set to any debug strings that the malware generates during runtime, encoded with base64.

Some of the debug strings include:

- We succ connected to ipc share
- WNetEnumResource failed with error %d
- file downloaded ok
- worm started

```
172.16.92.128
Array
(
    [id] => 49ac1a6e-cd89-4963-946b-1c8156b3cd8f
    [data] => d29ybSBzdGFydGVk
)
```

Figure 1 – Example request

These requests are sent to a single gate.php file with the path hardcoded in the spreader.

Spreader Changes

As stated above, the overall spreader functionality has not changed much. Instead, the authors have added in more verbose debugging, while also making the spreader more versatile in the payloads that it downloads. Additionally, the service name has changed in the newly updated spreader.

The only notable change to the spreader functionality is that if the spreader fails to brute-force the C$ share, the spreader will then attempt to brute-force the ADMIN$ share.

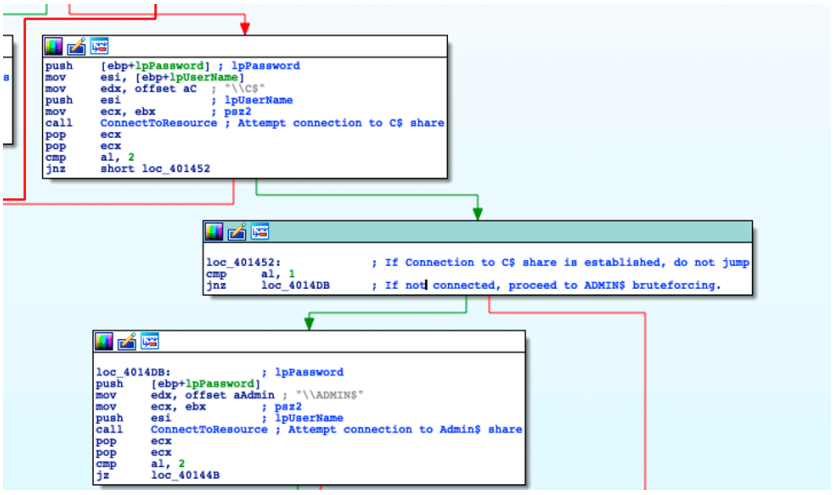

Figure 2 – Spreader bruteforcing code

Additionally, before the spreader attempts to brute-force C$/ADMIN$, it attempts to download, from a hardcoded IP, the service binary that it installs remotely. If this download fails, it sends the debug string "error downloading file" before quitting.

Service.exe Changes

Pulled down from a hardcoded URL, Service.exe is the executable used to install Emotet onto infected machines[CC1] . This binary, like the old Service.exe, will only detonate if first launched as a service. Unlike the old Service.exe however, the updated Service.exe downloads an Emotet binary from the C2 instead of containing a binary packaged inside of it.

Upon startup of Service.exe, the malware connects out to the same gate.php used by the spreader and sends the debug string "*remote service runned Downloading payload...*". Next, it attempts to connect to a hardcoded C2 where it pulls down the Emotet binary, saving the downloaded file as "firefox.exe."

After updating the C2 with the download status, if Emotet was successfully downloaded, Service.exe sends "*payload downloaded ok*" to the C2 before executing the dropped file.

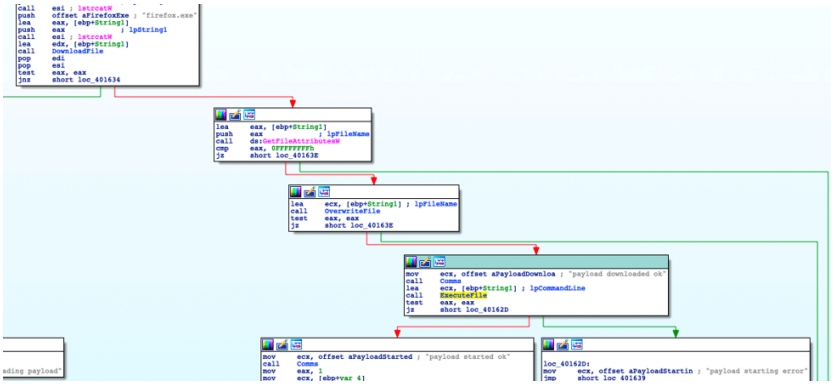

Figure 3 – Download and execute code

By downloading the Emotet loader directly from the C2, Service.exe can ensure that it has the most recent loader, without needing to package it inside itself. Additionally, this method helps to avoid detections that may flag off of the Emotet loader, but not the service executable.

Notable Artifacts

While analyzing the spreader/Service.exe combo, Binary Defense analysts uncovered some interesting and notable artifacts that lend some insight into the development process for the spreader. While looking at strings for the spreader executable, Binary Defense noticed that the hardcoded URL used by Service.exe to pull down the Emotet loader was also present in the spreader executable. Additionally, the drop name for the Emotet loader (firefox.exe) was also present. However, both were unused. This hints that it is possible that the spreader and service combo were once a single file.

```
wlan_notification_msm_adapter_removal
wlan_notification_msm_adapter_operation_mode_change
http://69.43.168.245/UUUU030G182K9N73VR35HW/service.exe
http://69.43.168.245/UUUU030G182K9N73VR35HW/em_wifi.exe
firefox.exe
```

Figure 4 – Strings

IOCs

Display Name: AsusService Service Name: ASUS system Service		
8a4239737f41b7f1730e6b6fdd2ecc3f1a4862bb6ab17f8a3d5eeba59423a8a0	69[.]43[.]168[.]245 /UUUU030G182K9N73VR35HW/service.exe /UUUU030G182K9N73VR35HW/gate.php	C:A %T
3c72f2fe57a0a6f1566bcc809b1039fafb483d5cb15efe8a03c3d68d5db2589f	69[.]43[.]168[.]245 /OWP3940LD8UWMAZ26XCSQWV182K9/service.exe /OWP3940LD8UWMAZ26XCSQWV182K9/gate.php	C:m %T

YARA rule

```
rule Emotet_WiFi_Spreader {
meta:
 title = "Emotet Wi-Fi Spreader identification"
 author = "<james.quinn@binarydefense.com>"
 strings:
// 00401174 83 c4 0c    ADD    ESP,0xc
//00401177 89 5d e4    MOV    dword ptr [EBP + local_20],EBX
//0040117a 8d 85 e0    LEA    EAX=>local_824,[EBP + 0xfffff7e0]
//     f7 ff ff
//00401180 89 5d f0    MOV    dword ptr [EBP + local_14],EBX
//00401183 89 45 f4    MOV    dword ptr [EBP + local_10],EAX
//00401186 8d 45 e0    LEA    EAX=>local_24,[EBP + -0x20]
//00401189 6a 01       PUSH    0x1
//0040118b ff 75 08    PUSH    dword ptr [EBP + param_3]
//0040118e ff 75 0c    PUSH    dword ptr [EBP + param_4]
//00401191 50       PUSH    EAX
//00401192 ff 15 64    CALL    dword ptr [
>MPR.DLL::WNetAddConnection2W]
//     01 41 00
//00401198 83 f8 35    CMP    EAX,0x35
//0040119b 75 0e       JNZ    LAB_004011ab


$WNetAddConnection2W = { 83 c4 0c 89 5d e4 8d 85 e0 f7 ff ff 89 5d f0 89 45
f4 8d 45 e0 6a 01 ff 75 08 ff 75 0c 50 ff 15 ?? ?? ?? ?? 83 f8 35 75 }
 $s1 = "We succ connected to ipc share"
condition:
 all of them
}
```

SURICATA rule:

```
alert tcp $HOME_NET any -> $EXTERNAL_NET [80,443,8080,7080,21,50000,995](msg:"BDS MALICIOUS Emotet
Worming Traffic Likely";content:"d29ybSBzdGFydGVk";content:"POST";http_method;classtype:spreader;sid:7;rev:1)
```
