Meet MyloBot - A New Highly Sophisticated Never-Seen-Before Botnet That's Out In The Wild

deepinstinct.com/2018/06/20/meet-mylobot-a-new-highly-sophisticated-never-seen-before-botnet-thats-out-in-the-wild/

June 20, 2018

Learn more

June 20, 2018 | Dalya Guttman

Over the past few years, we have seen various ways of spreading malicious code, one main infrastructure of spreading malware being the dark web.

Lately, we have noticed a highly complicated botnet (number of internet-connected devices, where the owner can control them using command and control servers), which was detected and prevented in one of our client's live environment and devices - by our deep learning cybersecurity solution. This botnet presents three different layers of evasion techniques, including usage of command and control servers to download the final payload. the combination and complexity of these techniques were never seen in the wild before.

Botnets can theoretically perform anything – depending on the payload. The payload can vary from DDoS attacks, steal data, and even installation ransomware which can cause tremendous damage.

What it does

This highly sophisticated botnet incorporates different malicious techniques:

- Anti VM techniques
- Anti-sandbox techniques
- Anti-debugging techniques
- Wrapping internal parts with an encrypted resource file
- Code injection
- Process hollowing - a technique where an attacker creates a new process in a suspended state, and replaces its image with the one that is to be hidden
- Reflective EXE - executing EXE files directly from memory, without having them on disk. This kind of reflection is not very common and was first published by Deep Instinct in Blackhat USA 2016
- It also has a delaying mechanism of 14 days before accessing its command and control servers.

The fact that everything takes place in memory (while executing the main business logic of the botnet in an external process using code injection) makes it even harder to detect and trace.

When we traced the command and control server we revealed that it was used by other malware campaigns as well which originated from the dark web.

The dark web plays a critical part in the spread of malware: Its rather simple accessibility of services and knowledge has made it easy for any attacker to gain much more abilities in minimum effort. The first example for this, is the shared knowledge in forums: in the dark web, attackers trade methods and techniques in underground forums, thus exposing knowledge to additional malware developers.

Another example, which has increased in the past couple of years, is the amount of malware for sale on dark web markets. By using the dark web, anyone today can access an online market and purchase a malware. Prices vary, from simple malware that costs several dollars to malware sold at hundreds of dollars as “fully undetectable”. Other than the malware itself, malware developers can purchase services that assist in the infection process. An attacker can purchase access to exploit kits, buy traffic of tens of thousands of users to a web page, or even buy a full ransomware-as-a-service for his own use.

Malware vs. Malware

Part of this malware process is terminating and deleting instances of other malware. It checks for known folders that malware “lives” in (“Application Data” folder), and if a certain file is running – it immediately terminates it and deletes its file. It even aims for specific folders of other botnets such as DorkBot.

We estimate this rare and unique behavior is because of money purposes within the Dark web. Attackers compete against each other to have as many “zombie computers” as possible in order to increase their value when proposing services to other attackers, especially when it comes to spreading infrastructures. The more computers – the more money an attacker can make. This is something we’re seeing here as well.

```
004044E5   FF50 14          CALL DWORD PTR DS:[EAX+14]              KERNEL32.Process32First
004044E8   EB 25            JMP SHORT 0040450F
004044EA   FFB424 3401000(  PUSH DWORD PTR SS:[ESP+134]
004044F1   8D4424 30        LEA EAX,[ESP+30]
[00413654]=754F4C51 (KERNEL32.Process32First)
```

Comparing current running process on the list to a file located in %APPDATA% (“LoadOrd.exe” in this case)

```
004044EA   FFB424 3401000(  PUSH DWORD PTR SS:[ESP+134]             PROCESSENTRY32.szExeFile
004044F1   8D4424 30        LEA EAX,[ESP+30]
004044F5   50               PUSH EAX                                ASCII "smss.exe"
004044F6   E8 ADCBFFFF      CALL CompareStrings
004044FB   59               POP ECX
004044FC   59               POP ECX
004044FD   85C0             TEST EAX,EAX
004044FF   74 14            JE SHORT 00404515
00404501   8D4424 08        LEA EAX,[ESP+8]
00404505   50               PUSH EAX
00404506   A1 C83C4100      MOV EAX,DWORD PTR DS:[413CC8]           ASCII "@6A"
0040450B   55               PUSH EBP
0040450C   FF50 18          CALL DWORD PTR DS:[EAX+18]              KERNEL32.Process32Next
0040450F   85C0             TEST EAX,EAX
Stack [0348F4D0]=0040450F (current registers)
EAX=0348F508, ASCII "smss.exe" (current registers)
```

Address	Value	Comments
0348F4D4	0348F508	ASCII "smss.exe"
0348F4D8	0348F74C	ASCII "LoadOrd.exe"

In case there is a match, terminate the process and delete it

```
0040452B   FF90 A0000000   CALL DWORD PTR DS:[EAX+0A0]          KERNEL32.OpenProcess
00404531   8BF8            MOV EDI,EAX
00404533   85FF            TEST EDI,EDI
00404535   74 37           JE SHORT 0040456E
00404537   56              PUSH ESI
00404538   57              PUSH EDI
00404539   FF15 14704000   CALL DWORD PTR DS:[407014]          KERNEL32.GetProcessId
0040453F   8BF0            MOV ESI,EAX
00404541   FF15 0C704000   CALL DWORD PTR DS:[40700C]          KERNEL32.GetCurrentProcessId
00404547   3BF0            CMP ESI,EAX
00404549   5E              POP ESI
0040454A   74 22           JE SHORT 0040456E
0040454C   6A 09           PUSH 9
0040454E   57              PUSH EDI
0040454F   FF15 10704000   CALL DWORD PTR DS:[407010]          KERNEL32.TerminateProcess
00404555   8B0D C83C4100   MOV ECX,DWORD PTR DS:[413CC8]       ASCII "@6A"
```

```
004048D5   FF15 34704000   CALL DWORD PTR DS:[407034]          KERNEL32.MoveFileA
004048DB   8D85 B8FCFFFF   LEA EAX,[EBP-348]
004048E1   50              PUSH EAX
[00407034]=754F568A (KERNEL32.MoveFileA)

Address    Value      Comments
0348F60C   0348F860   Existing = "C:\Users\Generic\AppData\Roaming\LoadOrd.exe"
0348F610   0348F61C   New = "C:\Users\Generic\AppData\Roaming\LoadOrd.exe.local.backup"
```

```
004048E2   FF15 28704000   CALL DWORD PTR DS:[407028]          KERNEL32.DeleteFileA
004048E8   8D85 BCFDFFFF   LEA EAX,[EBP-244]
[00407028]=754F7434 (KERNEL32.DeleteFileA) - jumps to KERNELBASE.DeleteFileA

Address    Value      Comments
0348F610   0348F61C   ASCII "C:\Users\Generic\AppData\Roaming\LoadOrd.exe.local.backup"
```

The Expected Damage

Once installed, the botnet shuts down Windows Defender and Windows Update while blocking additional ports on the Firewall. It also shuts down and deletes any EXE file running from %APPDATA% folder, which can cause loss of data. The main functionality of the botnet enables an attacker to take complete control of the user's system - it behaves as a gate to download additional payloads from the command and control servers. The expected damage here depends on the payload the attacker decides to distribute. It can vary from downloading and executing ransomware and banking trojans, among others. This can result in loss of tremendous amount of data, the need to shut down computers for recovery purposes, which can lead to disasters in enterprises. The fact that the botnet behaves as a gate for additional payloads, puts the enterprise in risk for leak of sensitive data as well, following the risk of keyloggers / banking trojans installations.

Although this kind of complexity in the malware's structure is extremely rare, Deep Instinct detected and prevented it on a client's production environment. Once again, this is a true testament for the superiority of deep learning based solutions in the cybersecurity warfare.

For a full detailed analysis of Mylobot, download our free research paper on botnets>>