CyberGate Threat Report

Date: 20/05/2021
Hussain Kathawala

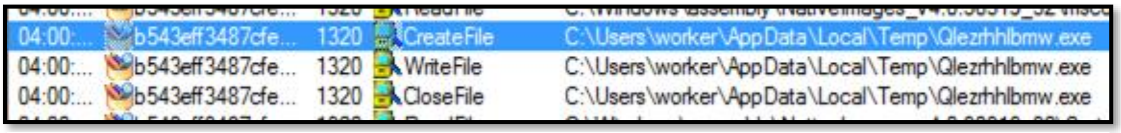
CyberGate is a Remote Access Trojan (RAT) that allows an attacker to gain unauthorized access to the victim's system. Attackers can remotely connect to the compromised system from anywhere around the world. The Malware author generally uses this program to steal private information like passwords, files, etc. It might also be used to install malicious software on the compromised systems.

Overview

The initial Malware is a dropper that executes and drops a malicious file into the victim system without the consent of the user. The dropped file starts its execution and performs anti-VM checks to prevent the execution in any virtual environment. After which the Malware performs various malicious activities, like Keylogging, and stealing sensitive information. Then it creates a legitimate process to communicate with the C2 server.

Infection

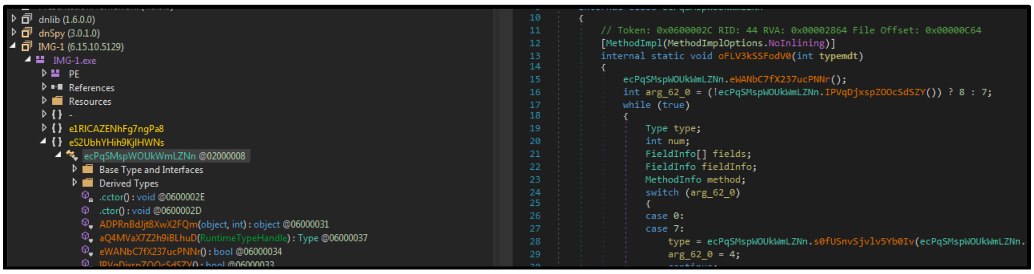
The initial executable is a PE32 file, for Intel architecture compiled with .NET. It drops another executable in the temp folder with the name "`Qlezrhhlbmw.exe`".



```
04:00:...  b543eff3487cfe...  1320  CreateFile  C:\Users\worker\AppData\Local\Temp\Qlezrhhlbmw.exe
04:00:...  b543eff3487cfe...  1320  WriteFile   C:\Users\worker\AppData\Local\Temp\Qlezrhhlbmw.exe
04:00:...  b543eff3487cfe...  1320  CloseFile   C:\Users\worker\AppData\Local\Temp\Qlezrhhlbmw.exe
```

Figure 1

The dropper when disassembled, shows classes with random generated names like "`ADPRnBdJjt8XwX2FQm`" and "`eWANbC7fX237ucPNNr`" to avoid detection and analysis.

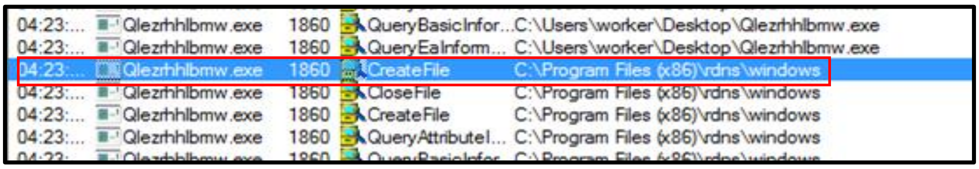


```
dnlib (1.6.0.0)
dnSpy (3.0.1.0)
IMG-1 (6.15.10.5129)
  IMG-1.exe
    PE
    References
    Resources
    { } -
    { } e1RICAZENhFg7ngPa8
    { } eS2UbhYHih9KjlHWNs
      ecPqSMspWOUkWmLZNn @02000008
        Base Type and Interfaces
        Derived Types
        .cctor() : void @0600002E
        .ctor() : void @0600002D
        ADPRnBdJjt8XwX2FQm(object, int) : object @06000031
        aQ4MVaX7Z2h9iBLhuD(RuntimeTypeHandle) : Type @06000037
        eWANbC7fX237ucPNNr() : bool @06000034
```

```
{
    // Token: 0x0600002C RID: 44 RVA: 0x00002864 File Offset: 0x00000C64
    [MethodImpl(MethodImplOptions.NoInlining)]
    internal static void oFLV3kSSFodV0(int typemdt)
    {
        ecPqSMspWOUkWmLZNn.eWANbC7fX237ucPNNr();
        int arg_62_0 = (!ecPqSMspWOUkWmLZNn.IPVqDjxspZOOcSdSZY()) ? 8 : 7;
        while (true)
        {
            Type type;
            int num;
            FieldInfo[] fields;
            FieldInfo fieldInfo;
            MethodInfo method;
            switch (arg_62_0)
            {
            case 0:
            case 7:
                type = ecPqSMspWOUkWmLZNn.s0fUSnvSjvlv5Yb0Iv(ecPqSMspWOUkWmLZNn.
                arg_62_0 = 4;
```

Figure 2

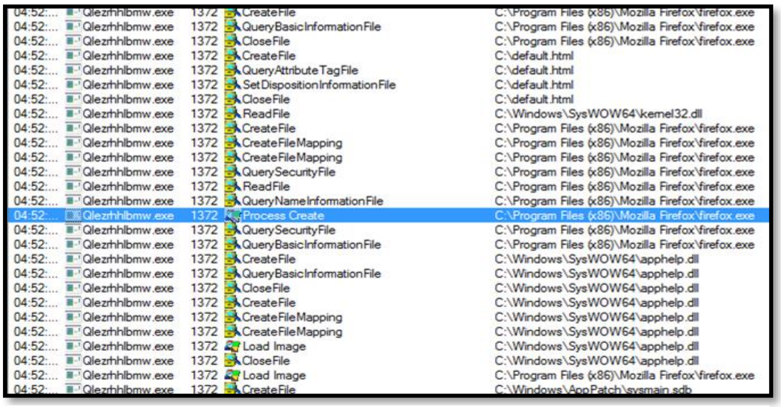
Once the actual RAT is dropped, its execution begins. To remain persistent in the system, the malware creates copies of itself in the folder "`c:\\programfiles(x86)\rdns`" as "`windows`" as shown in the figure below. The copy of the executable gets deleted after the execution gets completed.



```
04:23:...  Qlezrhhlbmw.exe  1860  QueryBasicInfor...  C:\Users\worker\Desktop\Qlezrhhlbmw.exe
04:23:...  Qlezrhhlbmw.exe  1860  QueryEaInform...    C:\Users\worker\Desktop\Qlezrhhlbmw.exe
04:23:...  Qlezrhhlbmw.exe  1860  CreateFile          C:\Program Files (x86)\rdns\windows
04:23:...  Qlezrhhlbmw.exe  1860  CloseFile           C:\Program Files (x86)\rdns\windows
04:23:...  Qlezrhhlbmw.exe  1860  CreateFile          C:\Program Files (x86)\rdns\windows
04:23:...  Qlezrhhlbmw.exe  1860  QueryAttributeI...  C:\Program Files (x86)\rdns\windows
```

Figure 3

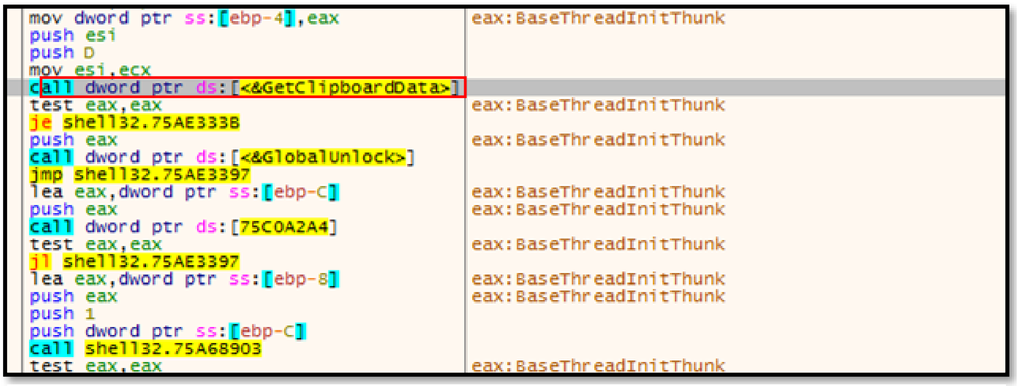
It creates a legitimate process in firefox.exe and it injects a code using various functions as seen in the below figure. It allocates memory through this step.



```
04:52:...  Qlezrhhlbmw.exe  1372  CreateFile                    C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  QueryBasicInformationFile     C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  CloseFile                     C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  CreateFile                    C:\default.html
04:52:...  Qlezrhhlbmw.exe  1372  QueryAttributeTagFile         C:\default.html
04:52:...  Qlezrhhlbmw.exe  1372  SetDispositionInformationFile C:\default.html
04:52:...  Qlezrhhlbmw.exe  1372  CloseFile                     C:\default.html
04:52:...  Qlezrhhlbmw.exe  1372  ReadFile                      C:\Windows\SysWOW64\kernel32.dll
04:52:...  Qlezrhhlbmw.exe  1372  CreateFile                    C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  CreateFileMapping             C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  CreateFileMapping             C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  QuerySecurityFile             C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  ReadFile                      C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  QueryNameInformationFile      C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  Process Create                C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  QuerySecurityFile             C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  QueryBasicInformationFile     C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  CreateFile                    C:\Windows\SysWOW64\apphelp.dll
04:52:...  Qlezrhhlbmw.exe  1372  QueryBasicInformationFile     C:\Windows\SysWOW64\apphelp.dll
04:52:...  Qlezrhhlbmw.exe  1372  CloseFile                     C:\Windows\SysWOW64\apphelp.dll
04:52:...  Qlezrhhlbmw.exe  1372  CreateFile                    C:\Windows\SysWOW64\apphelp.dll
04:52:...  Qlezrhhlbmw.exe  1372  CreateFileMapping             C:\Windows\SysWOW64\apphelp.dll
04:52:...  Qlezrhhlbmw.exe  1372  CreateFileMapping             C:\Windows\SysWOW64\apphelp.dll
04:52:...  Qlezrhhlbmw.exe  1372  Load Image                    C:\Windows\SysWOW64\apphelp.dll
04:52:...  Qlezrhhlbmw.exe  1372  CloseFile                     C:\Windows\SysWOW64\apphelp.dll
04:52:...  Qlezrhhlbmw.exe  1372  Load Image                    C:\Program Files (x86)\Mozilla Firefox\firefox.exe
04:52:...  Qlezrhhlbmw.exe  1372  CreateFile                    C:\Windows\AppPatch\sysmain.sdb
```

Figure 4

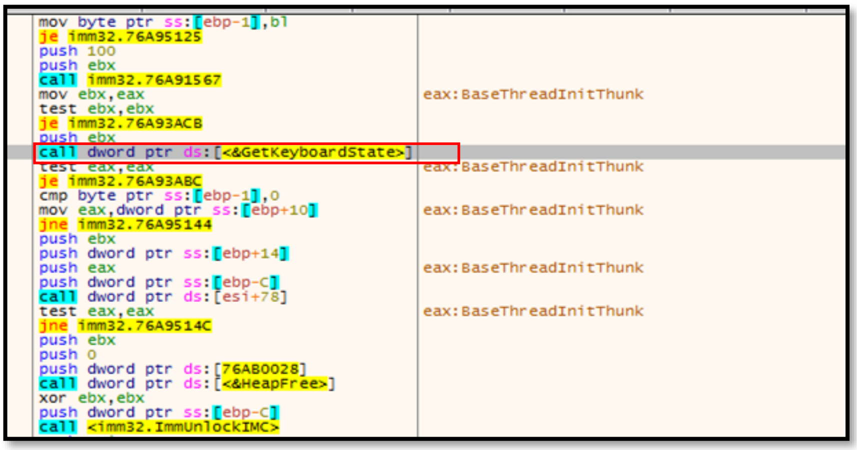
The malware contains functionality to read the clipboard data with the "GetClipboardData" function as seen below. The return value of this function is the handle to a clipboard object.



```
mov dword ptr ss:[ebp-4],eax               eax:BaseThreadInitThunk
push esi
push D
mov esi,ecx
call dword ptr ds:[<&GetClipboardData>]
test eax,eax                               eax:BaseThreadInitThunk
je shell32.75AE333B
push eax                                   eax:BaseThreadInitThunk
call dword ptr ds:[<&GlobalUnlock>]
jmp shell32.75AE3397
lea eax,dword ptr ss:[ebp-C]               eax:BaseThreadInitThunk
push eax                                   eax:BaseThreadInitThunk
call dword ptr ds:[75C0A2A4]
test eax,eax                               eax:BaseThreadInitThunk
jl shell32.75AE3397
lea eax,dword ptr ss:[ebp-8]               eax:BaseThreadInitThunk
push eax                                   eax:BaseThreadInitThunk
push 1
push dword ptr ss:[ebp-C]
call shell32.75A68903
test eax,eax                               eax:BaseThreadInitThunk
```

Figure 5

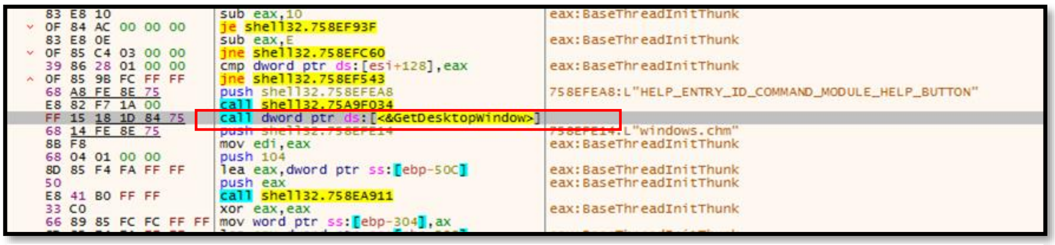
It also retrieves information about pressed keystrokes and acts as a Keylogger using the function "GetKeyboardState". It gives the status of the 256-virtual keys.



```
mov byte ptr ss:[ebp-1],bl
je imm32.76A95125
push 100
push ebx
call imm32.76A91567
mov ebx,eax                                eax:BaseThreadInitThunk
test ebx,ebx
je imm32.76A93ACB
push ebx
call dword ptr ds:[<&GetKeyboardState>]
test eax,eax                               eax:BaseThreadInitThunk
je imm32.76A93ABC
cmp byte ptr ss:[ebp-1],0
mov eax,dword ptr ss:[ebp+10]              eax:BaseThreadInitThunk
jne imm32.76A95144
push ebx
push dword ptr ss:[ebp+14]
push eax                                   eax:BaseThreadInitThunk
push dword ptr ss:[ebp-C]
call dword ptr ds:[esi+78]
test eax,eax                               eax:BaseThreadInitThunk
jne imm32.76A9514C
push ebx
push 0
push dword ptr ds:[76AB0028]
call dword ptr ds:[<&HeapFree>]
xor ebx,ebx
push dword ptr ss:[ebp-C]
call <imm32.ImmUnlockIMC>
```

Figure 6

The malware also has a functionality to retrieve handle of the desktop window of the victim system using the functionality, “`GetDesktopWindow`”.

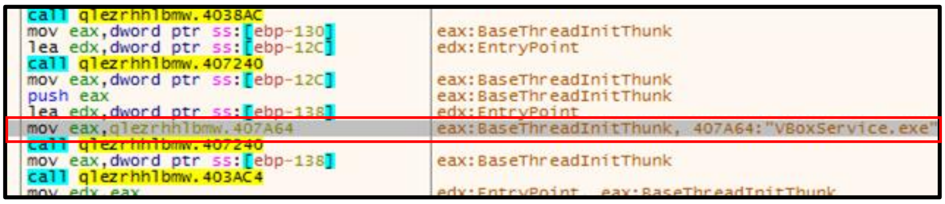


```
83 E8 10              sub eax,10                                 eax:BaseThreadInitThunk
0F 84 AC 00 00 00     je shell32.758EF93F
83 E8 0E              sub eax,E                                  eax:BaseThreadInitThunk
0F 85 C4 03 00 00     jne shell32.758EFC60
39 86 28 01 00 00     cmp dword ptr ds:[esi+128],eax             eax:BaseThreadInitThunk
0F 85 9B FC FF FF     jne shell32.758EF543
68 A8 FE 8E 75        push shell32.758EFEA8                      758EFEA8:L"HELP_ENTRY_ID_COMMAND_MODULE_HELP_BUTTON"
E8 82 F7 1A 00        call shell32.75A9F034
FF 15 18 1D 84 75     call dword ptr ds:[<&GetDesktopWindow>]
68 14 FE 8E 75        push shell32.758EFE14                      758EFE14:L"windows.chm"
8B F8                 mov edi,eax                                eax:BaseThreadInitThunk
68 04 01 00 00        push 104
8D 85 F4 FA FF FF     lea eax,dword ptr ss:[ebp-50C]             eax:BaseThreadInitThunk
50                    push eax                                   eax:BaseThreadInitThunk
E8 41 B0 FF FF        call shell32.758EA911
33 C0                 xor eax,eax                                eax:BaseThreadInitThunk
66 89 85 FC FC FF FF  mov word ptr ss:[ebp-304],ax
```

Figure 7

Anti-VM & Anti-Debug Feature

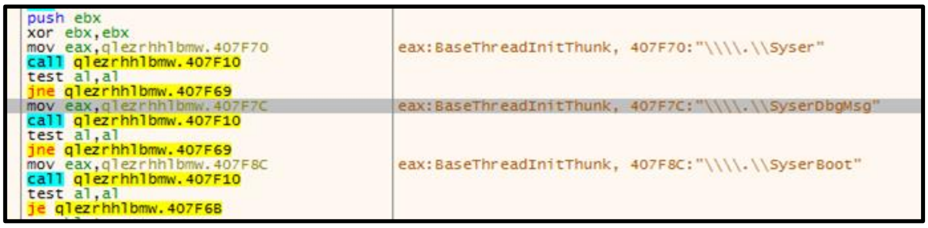
The malware performs various anti-VM checks on the victim machine. This statement is supported with the figure shown below.



```
call qlezrhhlbmw.4038AC
mov eax,dword ptr ss:[ebp-130]          eax:BaseThreadInitThunk
lea edx,dword ptr ss:[ebp-12C]          edx:EntryPoint
call qlezrhhlbmw.407240
mov eax,dword ptr ss:[ebp-12C]          eax:BaseThreadInitThunk
push eax                                eax:BaseThreadInitThunk
lea edx,dword ptr ss:[ebp-138]          edx:EntryPoint
mov eax,qlezrhhlbmw.407A64              eax:BaseThreadInitThunk, 407A64:"VBoxService.exe"
call qlezrhhlbmw.407240
mov eax,dword ptr ss:[ebp-138]          eax:BaseThreadInitThunk
call qlezrhhlbmw.403AC4
mov edx,eax                             edx:EntryPoint, eax:BaseThreadInitThunk
```

Figure 8

It also checks for the presence of kernel debugger in the system. This can be observed because of the strings like “`\\\\.\\Syser`”, “`\\\\.\\SyserDbgMsg`” and “`\\\\.\\SyserBoot`”.

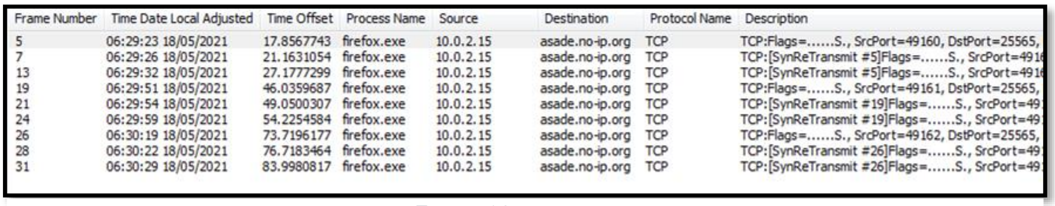


```
push ebx
xor ebx,ebx
mov eax,qlezrhhlbmw.407F70              eax:BaseThreadInitThunk, 407F70:"\\\\.\\Syser"
call qlezrhhlbmw.407F10
test al,al
jne qlezrhhlbmw.407F69
mov eax,qlezrhhlbmw.407F7C              eax:BaseThreadInitThunk, 407F7C:"\\\\.\\SyserDbgMsg"
call qlezrhhlbmw.407F10
test al,al
jne qlezrhhlbmw.407F69
mov eax,qlezrhhlbmw.407F8C              eax:BaseThreadInitThunk, 407F8C:"\\\\.\\SyserBoot"
call qlezrhhlbmw.407F10
test al,al
je qlezrhhlbmw.407F6B
```

Figure 9

Network Traffic Analysis

Once it captures the information, the malware tries to communicate with the C2 server “`aside.no-ip.org`” using the foreign process i.e., firefox.exe as we can see in the below figure.



Frame Number	Time Date Local Adjusted	Time Offset	Process Name	Source	Destination	Protocol Name	Description
5	06:29:23 18/05/2021	17.8567743	firefox.exe	10.0.2.15	asade.no-ip.org	TCP	TCP:Flags=......S., SrcPort=49160, DstPort=25565,
7	06:29:26 18/05/2021	21.1631054	firefox.exe	10.0.2.15	asade.no-ip.org	TCP	TCP:[SynReTransmit #5]Flags=......S., SrcPort=4916
13	06:29:32 18/05/2021	27.1777299	firefox.exe	10.0.2.15	asade.no-ip.org	TCP	TCP:[SynReTransmit #5]Flags=......S., SrcPort=4916
19	06:29:51 18/05/2021	46.0359687	firefox.exe	10.0.2.15	asade.no-ip.org	TCP	TCP:Flags=......S., SrcPort=49161, DstPort=25565,
21	06:29:54 18/05/2021	49.0500307	firefox.exe	10.0.2.15	asade.no-ip.org	TCP	TCP:[SynReTransmit #19]Flags=......S., SrcPort=49
24	06:29:59 18/05/2021	54.2254584	firefox.exe	10.0.2.15	asade.no-ip.org	TCP	TCP:[SynReTransmit #19]Flags=......S., SrcPort=49
26	06:30:19 18/05/2021	73.7196177	firefox.exe	10.0.2.15	asade.no-ip.org	TCP	TCP:Flags=......S., SrcPort=49162, DstPort=25565,
28	06:30:22 18/05/2021	76.7183464	firefox.exe	10.0.2.15	asade.no-ip.org	TCP	TCP:[SynReTransmit #26]Flags=......S., SrcPort=49
31	06:30:29 18/05/2021	83.9980817	firefox.exe	10.0.2.15	asade.no-ip.org	TCP	TCP:[SynReTransmit #26]Flags=......S., SrcPort=49

Figure 10

MITRE Attack Techniques Used

Technique ID	Technique
T1497	Virtualization/Sandbox Evasion
T1055	Process Injection
T1056	Input Capture
T1115	Clipboard Data
T1113	Screen Capture
T1036	Masquerading

IOC’s

4df346a12ef5679ec0b960d037c8f52a
2cad1ad59e145139cbab70260b1a2f19
hxxp://asade.no-ip.org
178.206.211.67

Subex Secure Protection

Subex Secure detects the sample as “SS_AI_Trojan_PE”.

Our Honeypot Network

This report has been prepared from the threat intelligence gathered by our Honeypot network. This Honeypot network is today operational in 62 cities across the world. These cities have at least one of the following attributes:

- Are landing centers for submarine cables
- Are internet traffic hotspots
- House multiple IoT projects with a high number of connected endpoints
- House multiple connected critical infrastructure projects
- Have academic and research centers focusing on IoT
- Have the potential to host multiple IoT projects across domains in the future.

Over 3.5 million attacks a day is being registered across this network of individual Honeypots. These attacks are studied, analysed, categorized, and marked according to a threat rank index, a priority assessment framework that we have developed within Subex. The Honeypot network includes over 4000 physical and virtual devices covering over 400 device architectures and varied connectivity mediums globally. These devices are grouped based on the sectors they belong to for purposes of understanding sectoral attacks. Thus, a layered flow of threat intelligence is made possible.