SECTRIO

MALWARE REPORT

Ipamor

Date: 28/09/2020

Shyava Tripathi,

Shikha Sangwan

Although a grizzled and slowed infection vector, file infectors utilising the advantage of hybrid payloads and cross-breeding propagation strategies are still prevalent in the wild. Ipamor, an old file infector, now embracing backdoor and information theft routines, has been able to persist over the years. It prepends its malicious code to executable files to keep the hoarde in motion but it ultimately establishes a backdoor and exfiltrates data to attacker controlled servers.

Ipamor is a prepending virus; the payload is added at the beginning of the infected file. When the infected file is executed, it extracts and drops the main payload file to the system and achieves persistance by adding its value to the registry keys. Running in the background, the virus searches for PE executable files and infects them along with opening backdoor access ports that allows the attacker to manipulate the infected systems.

INFECTION CHAIN

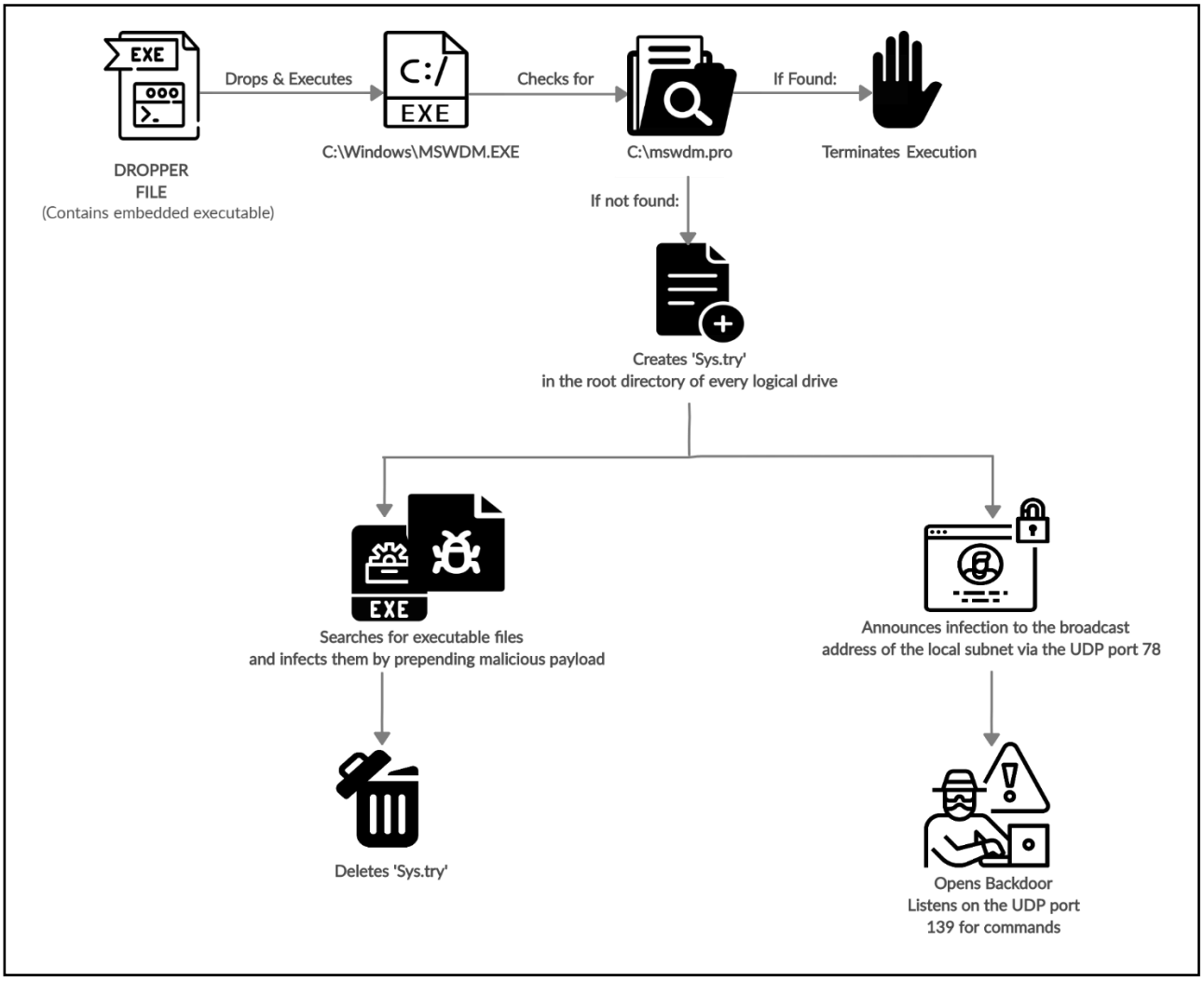

Figure 1: Ipamor Infection Chain

The infected executable file, acting as the dropper, contains the main payload in the form of an embedded executable file, 'mswdm.exe'. When the infected dropper file is executed, the payload 'mswdm.exe' is extracted and copied to 'C:\Windows' directory. Upon execution, it checks for a file by the name of 'mswdm.pro' in the systems root drive (C:\). If the said file is found, the malware does not infect any files and terminates itself. If the file is not detected, the infector attempts to infect files present in the root directory of every logical drive in the system by first creating a file

'sys.try' in the working directory followed by searching for and infecting files with the .EXE extension name (Figure 1). Once infected, the file 'sys.try' is deleted from the directory.

The infector, announces the infection to the local broadcast subnet via UDP port 78 and opens a backdoor on UDP port 139. It exfiltrates system information such as the computer name, architecture, processor details to the server and also allows the attacker to take screenshots, gather sensitive information and shutdown the infected computer.

ANALYSIS OVERVIEW

Upon execution, this virus creates a copy of itself in the Windows folder as MSWDM.EXE. Once created, the malware tries to achieve persistence by adding its value to the Windows Run registry key (Figure 2).

```
.text:0040127D
.text:0040127D                 push    ecx
.text:0040127E                 push    ecx
.text:0040127F                 push    ebx
.text:00401280                 push    ebp
.text:00401281                 push    esi
.text:00401282                 lea     eax, [esp+14h+phkResult]
.text:00401286                 push    edi
.text:00401287                 mov     esi, offset SubKey ; "Software\\Microsoft\\Windows\\CurrentVe"...
.text:0040128C                 push    eax             ; phkResult
.text:0040128D                 mov     edi, 80000002h
.text:00401292                 push    esi             ; lpSubKey
.text:00401293                 push    edi             ; hKey
.text:00401294                 call    ds:RegOpenKeyA
.text:0040129A                 test    eax, eax
.text:0040129C                 jz      short loc_4012AF
.text:0040129E                 lea     eax, [esp+18h+phkResult]
.text:004012A2                 push    eax             ; phkResult
.text:004012A3                 push    esi             ; lpSubKey
.text:004012A4                 push    edi             ; hKey
.text:004012A5                 call    ds:RegCreateKeyA
.text:004012AB                 test    eax, eax
.text:004012AD                 jnz     short loc_40132D
.text:004012AF
.text:004012AF loc_4012AF:                             ; CODE XREF: sub_40127D+1F↑j
.text:004012AF                 mov     esi, offset Data ; "MSWDM.EXE"
.text:004012B4                 push    esi             ; char *
.text:004012B5                 call    _strlen
.text:004012BA                 pop     ecx

; CHAR SubKey[]
SubKey          db 'Software\Microsoft\Windows\CurrentVersion\Run',0
                                                       ; DATA XREF: sub_40127D+A↑o
```

Figure 2: Achieving Persistence

Ipamor also modifies firewall-based registry keys to bypass firewall policies:

```
"v2.10|Action=Allow|Active=TRUE|Dir=In|Protocol=6|Profile=Public|App=C:\windows\mswdm.exe|Name=MSWDM|Desc=MSWDM|Defer=User|"
```

Upon achieving persistence, the malware searches for 'MSWDM.PRO' in the root drive of the system (Figure 3). If the file is encountered, the infection is not carried out and the malware stops execution. If the file is not found in the system, the malware attempts to infect the files.

The infection begins by creating a 'sys.try' file in root directory of the logical drive that the malware is trying to infect. Upon creation of 'sys.try', a file 'die<4 digit random number>.tmp' is created in the Temp directory to help with writing the malicious code to infected files (Figure 5). Ipamor then searches for files with the .EXE extension name (Figure 6) and infects them.

It attaches its code at the beginning and also adds extra 18 bytes of data at the end of the infected host files. The infected files may have an increase of roughly 130,000 to 140,000 bytes in the size of the file.

```
.text:004033F7                  push    ebp
.text:004033F8                  mov     ebp, esp
.text:004033FA                  sub     esp, 34Ch
.text:00403400                  push    ebx
.text:00403401                  push    esi
.text:00403402                  xor     ebx, ebx
.text:00403404                  push    edi
.text:00403405                  push    ebx             ; hTemplateFile
.text:00403406                  push    80h ; '€'       ; dwFlagsAndAttributes
.text:0040340B                  push    3               ; dwCreationDisposition
.text:0040340D                  push    ebx             ; lpSecurityAttributes
.text:0040340E                  push    1               ; dwShareMode
.text:00403410                  push    80000000h       ; dwDesiredAccess
.text:00403415                  push    offset FileName ; "c:\\mswdm.pro"
.text:0040341A                  call    ds:CreateFileA
.text:00403420                  cmp     eax, 0FFFFFFFFh
.text:00403423                  jz      short loc_403435
.text:00403425                  push    eax             ; hObject
.text:00403426                  call    ds:CloseHandle
.text:0040342C                  pop     edi
.text:0040342D                  pop     esi
.text:0040342E                  xor     eax, eax
.text:00403430                  pop     ebx
.text:00403431                  leave
.text:00403432                  retn    4
```

Figure 3: Malware searches for 'mswdm.pro'

```
.text:0040356E                  push    ebp
.text:0040356F                  mov     ebp, esp
.text:00403571                  sub     esp, 104h
.text:00403577                  push    104h            ; size_t
.text:0040357C                  lea     eax, [ebp+FileName]
.text:00403582                  push    0               ; int
.text:00403584                  push    eax             ; void *
.text:00403585                  call    _memset
.text:0040358A                  push    [ebp+arg_0]     ; char *
.text:0040358D                  lea     eax, [ebp+FileName]
.text:00403593                  push    eax             ; char *
.text:00403594                  call    _strcat
.text:00403599                  lea     eax, [ebp+FileName]
.text:0040359F                  push    offset aSysTry  ; "sys.try"
.text:004035A4                  push    eax             ; char *
.text:004035A5                  call    _strcat
.text:004035AA                  add     esp, 1Ch
.text:004035AD                  lea     eax, [ebp+FileName]
.text:004035B3                  push    0               ; hTemplateFile
.text:004035B5                  push    80h ; '€'       ; dwFlagsAndAttributes
.text:004035BA                  push    2               ; dwCreationDisposition
.text:004035BC                  push    0               ; lpSecurityAttributes
.text:004035BE                  push    1               ; dwShareMode
.text:004035C0                  push    0C0000000h      ; dwDesiredAccess
.text:004035C5                  push    eax             ; lpFileName
.text:004035C6                  call    ds:CreateFileA
.text:004035CC                  cmp     eax, 0FFFFFFFFh
.text:004035CF                  jnz     short loc_4035D5
.text:004035D1                  xor     eax, eax
```

Figure 4: Malware creates sys.try

```
.text:00403483          call    ds:GetWindowsDirectoryA
.text:00403489          push    edi             ; lpTempFileName
.text:0040348A          push    ebx             ; uUnique
.text:0040348B          lea     eax, [ebp+Buffer]
.text:00403491          push    offset aDie     ; "die"
.text:00403496          push    eax             ; lpPathName
.text:00403497          call    ds:GetTempFileNameA
.text:0040349D          mov     [ebp+var_8], ebx
.text:004034A0          lea     edi, [ebp+var_144]
.text:004034A6          mov     [ebp+var_4], 1Ah
```

Figure 5: Die<4 digit number>.tmp is created

```
.text:004036E0          mov     ebp, esp
.text:004036E2          sub     esp, 4C4h
.text:004036E8          push    ebx
.text:004036E9          push    esi
.text:004036EA          push    edi
.text:004036EB          xor     esi, esi
.text:004036ED          push    104h            ; size_t
.text:004036F2          lea     eax, [ebp+FileName]
.text:004036F8          push    esi             ; int
.text:004036F9          push    eax             ; void *
.text:004036FA          mov     [ebp+var_4], esi
.text:004036FD          call    _memset
.text:00403702          push    [ebp+arg_0]     ; char *
.text:00403705          lea     eax, [ebp+FileName]
.text:0040370B          push    eax             ; char *
.text:0040370C          call    _strcat
.text:00403711          lea     eax, [ebp+FileName]
.text:00403717          push    offset aExe     ; "\\*.exe"
.text:0040371C          push    eax             ; char *
.text:0040371D          call    _strcat
.text:00403722          add     esp, 1Ch
.text:00403725          lea     eax, [ebp+FindFileData]
.text:0040372B          push    eax             ; lpFindFileData
.text:0040372C          lea     eax, [ebp+FileName]
.text:00403732          push    eax             ; lpFileName
.text:00403733          call    ds:FindFirstFileA
.text:00403739          cmp     eax, 0FFFFFFFFh
.text:0040373C          mov     [ebp+hFindFile], eax
.text:0040373F          jz      loc_4039C0
.text:00403745          mov     ebx, offset ExistingFileName
.text:0040374A          jmp     short loc_40374E
```

Figure 6: Ipamor searches for files with .exe extension to infect

Ipamor broadcasts its infection to local subnet via the UDP port 78 (Figure 7).

```
669 136.466273100   10.10.10.15   10.10.10.255     UDP   53 139 → 78 Len=11
670 136.466822600   10.10.10.15   10.10.255.255    UDP   53 139 → 78 Len=11
671 136.467133700   10.10.10.15   10.255.255.255   UDP   53 139 → 78 Len=11
```

Figure 7: Ipamor network capture

Ipamor exfiltrates computer name to the attacker upon broadcasting the infection. The character 'g' is prepended and '_V' is appended to the computer name while sending (Figure 8a, 8b).

```
> Frame 671: 53 bytes on wire (424 bits), 53 bytes captured (424 bits)
> Ethernet II, Src: PcsCompu_21:1f:f5 (08:00:27:21:1f:f5), Dst: Fortinet_ff:a5:0b (70:4c:a5:ff:a5:0b)
> Internet Protocol Version 4, Src: 10.10.10.15, Dst: 10.255.255.255
v User Datagram Protocol, Src Port: 139, Dst Port: 78
      Source Port: 139
      Destination Port: 78
      Length: 19
      Checksum: 0x1f3c [unverified]
      [Checksum Status: Unverified]
      [Stream index: 45]
    > [Timestamps]
v Data (11 bytes)
      Data: 675741525a4f4e45315f56
      [Length: 11]
```

```
0000  70 4c a5 ff a5 0b 08 00  27 21 1f f5 08 00 45 00   pL······ '!····E·
0010  00 27 01 c9 00 00 80 11  00 00 0a 0a 0a 0f 0a ff   ·'······ ········
0020  ff ff 00 8b 00 4e 00 13  1f 3c 67 57 41 52 5a 4f   ·····N·· ·<gWARZO
0030  4e 45 31 5f 56                                    NE1_V
```

```
.text:00402694                 push    ebp
.text:00402695                 mov     ebp, esp
.text:00402697                 sub     esp, 154h
.text:0040269D                 push    ebx
.text:0040269E                 push    esi
.text:0040269F                 push    edi
.text:004026A0                 xor     ebx, ebx
.text:004026A2                 push    40h ; '@'       ; size_t
.text:004026A4                 lea     eax, [ebp+cp]
.text:004026A7                 push    ebx             ; int
.text:004026A8                 push    eax             ; void *
.text:004026A9                 call    _memset
.text:004026AE                 mov     eax, 0FFh
.text:004026B3                 mov     [ebp+buf], 67h ; 'g'
.text:004026BA                 mov     [ebp+nSize], eax
.text:004026BD                 push    eax             ; size_t
.text:004026BE                 lea     eax, [ebp+Buffer]
.text:004026C4                 push    ebx             ; int
.text:004026C5                 push    eax             ; void *
.text:004026C6                 call    _memset
.text:004026CB                 add     esp, 18h
.text:004026CE                 lea     eax, [ebp+nSize]
.text:004026D1                 push    eax             ; nSize
.text:004026D2                 lea     eax, [ebp+Buffer]
.text:004026D8                 push    eax             ; lpBuffer
.text:004026D9                 call    ds:GetComputerNameA
.text:004026DF                 mov     esi, offset aV  ; "_V"
.text:004026E4                 lea     eax, [ebp+Buffer]
.text:004026EA                 push    esi             ; char *
.text:004026EB                 push    eax             ; char *
.text:004026EC                 call    _strcat
.text:004026F1                 push    esi             ; char *
.text:004026F2                 call    _strlen
```

Figure 8a, 8b: Ipamor exfiltrating Computer name and broadcasting infection

Ipamor creates a socket and establishes a backdoor on UDP port 139 (Figure 9). Once established, Ipamor uses this backdoor to send victim desktop window data and screenshots to the attacker.

```
.text:00402A71                  mov     [ebp+namelen], edi
.text:00402A74                  call    ds:getsockname
.text:00402A7A                  push    40h ; '@'       ; size_t
.text:00402A7C                  lea     eax, [ebp+buf]
.text:00402A7F                  push    esi             ; int
.text:00402A80                  push    eax             ; void *
.text:00402A81                  call    _memset
.text:00402A86                  add     esp, 0Ch
.text:00402A89                  mov     [ebp+buf], 21h ; '!'
.text:00402A8D                  push    dword ptr [ebp+name.sa_data] ; netshort
.text:00402A90                  call    ds:ntohs
.text:00402A96                  movzx   eax, ax
.text:00402A99                  mov     [ebp+var_1C], eax
.text:00402A9C                  lea     eax, [ebp+var_1C]
.text:00402A9F                  push    4               ; size_t
.text:00402AA1                  push    eax             ; void *
.text:00402AA2                  lea     eax, [ebp+var_77]
.text:00402AA5                  push    eax             ; void *
.text:00402AA6                  call    _memcpy
.text:00402AAB                  add     esp, 0Ch
.text:00402AAE                  lea     eax, [ebp+to]
.text:00402AB1                  push    edi             ; tolen
.text:00402AB2                  push    eax             ; to
.text:00402AB3                  push    esi             ; flags
.text:00402AB4                  lea     eax, [ebp+buf]
.text:00402AB7                  push    5               ; len
.text:00402AB9                  push    eax             ; buf
.text:00402ABA                  push    [ebp+s]         ; s
.text:00402ABD                  call    ds:sendto
.text:00402AC3                  push    5               ; backlog
.text:00402AC5                  push    [ebp+var_4]     ; s
.text:00402AC8                  call    ds:listen
.text:00402ACE                  test    eax, eax
.text:00402AD0                  jnz     loc_402C66
.text:00402AD6                  lea     eax, [ebp+timeout]
.text:00402AD9                  mov     [ebp+addrlen], edi
.text:00402ADC                  mov     edi, [ebp+var_4]
```

Figure 9: Ipamor establishes a backdoor on UDP port 139

Ipamor also contains capabilities to shut down the infected device and terminate firewall and antivirus (NORTON) processes for smooth execution.

INDICATORS OF COMPROMISE

Sample Hashes:

ff8b9ce5209f0ed9a7fb4a3ef12bedb6	674c03e9ea554e75c59f356e8db0855e
1ff48dffd35997b36e3b6621655128c1	b4fb541bfa77b2ab82e3738090946a99
f98f12dbb3e47fc5caac72bdb4f6c673	52472f5c57aac8198ccd7393e3b984bd
789fd1be74e066598339086cb7cc1465	8ce1ee1bba0513ec859411e4d7df5042
caac776f52da2abe40901c51537ac7c1	947818cd298167732b836fc1f044ba77
4ee117ee3cd9c4f8f6743219c3688e14	03c884e7680fe5f8a89405a47a2692e4
f6fae487f7e4a76c72d64a41a6188d73	1e412717544171f1c85932c1b2915f81
7d9869dec5925df51d95fcbbf7fa2147	c6430746ce048d024aaa788c5e3004a4
9cde162739bc1a3633d67504eb25c316	d3f6f4009a729a92615b4c22f66f4ab5
e08bc470647c21bd4553ad75c67cdad3	

Sample Attribute IOCs:

File Size	436.24 KB	
Compilation Time	11-07-2002	
Entry Point	0x11001	
No. of Sections	6	
Entropy of Sections	.text .rdata .data .rsrc	7.97 7.9 7.63 2.37
Chi2 value of sections	.text .rdata .data .rsrc	1476.69 271.25 3651 69417
XOR Key	96 3A 6E CA	
Code Size	40960	

Suspicious Strings:

c:\mswdm.pro	XXXXXXXXXXXXXXXXXXXXXXXX.EXE
sys.try	IPARMOR
\welcome.exe	TROJAN
\system\kernel32.exe	Mswdm.exe

MITRE ATTACK MATRIX

TACTIC	ID	NAME	DESCRIPTION
COMMAND AND CONTROL	T1105	Ingress Tool Transfer	It receives and writes data from server to client. It also transfers file named sys.try to the root directory of every logical drive present in the system.
	T1095	Non-Application Layer Protocol	It uses UDP for C2 communications. It announces the infection via the UDP port 78 and listens on the UDP port 139 for commands.
DISCOVERY	T1083	File and directory discovery	It checks for files with .exe extensions and also discovers root directories.
	T1082	System Information Discovery	It discovers system information like system manufacturer, system product name, BIOS information and system volume information and also queries for CPU information (cpuid), disk information and locales information (e.g. system language)

	T1057	Process Discovery	It queries a list of all running processes and may try to detect the windows explorer processes which it uses for injection.
	T1124	System Time Discovery	It queries local time and current time zone information.
	T1518.001	Security Software Discovery	It checks if any anitvirus or firewall program is installed via WMI. It also contains functionality to check if a debugger is running i.e. IsDebuggerPresent.
EXECUTION	T1129	Shared Modules	It uses local DLLs to execute its malicious payloads.
	T1106	Native API	It has an evasive block which contains many API calls or an API chain which may stop execution after checking a module file name. It also has functions to dynamically determine API calls.
IMPACT	T1529	System Shutdown/Reboot	It takes access to system startup which may include commands to shutdown or reboot the system.
PERSISTENCE	T1547.001	Boot or Logon Autostart Execution::Registry Run Keys / Startup Folder	It modifies auto-execute functionality by creating a value in the following registry paths: HKLM\SOFTWARE\WOW6432NODE\MICROSOFT\WINDOWS\CURRENTVERSION\RUN HKLM\SOFTWARE\WOW6432NODE\MICROSOFT\WINDOWS\CURRENTVERSION\RUNSERVICES Which controls automatic startup of services during boot.
PRIVILEGE ESCALATION	T1134	Access Token Manipulation	It contains functionality to launch a process as a different user.
	T1055	Process Injection	It tries to detect the windows explorer process and injects code into them.
DEFENCE EVASION	T1036	Masquerading	Ipamor contains invalid code signatures to mimic legitimate files.
	T1070.004	File Deletion	It deletes the main dropper file after execution. It also deletes 'sys.try' from the directory it infects.
	T1112	Modify Registry	It sets new registry values to bypass firewall policies.
	T1562.001	Impair Defences: Disable or Modify Tools	It terminates the following active processes related to firewall and antivirus programs: FIREWALL, NORTON
	T1497	Virtualization or Sandbox Evasion	It checks if the current process is being debugged and queries disk information

			which is often used to detect virtual machines.
	T1140	Deobfuscate/Decode Files or Information	Ipamor uses 'ASPACK' for obfuscation and also has the ability to encode the data using XOR keys.

Sectrio Protection

Sectrio detects Ipamor infector as 'SS_Gen_Ipamor_PE_A' and 'SS_Gen_Ipamor_PE_B'.

Our Honeypot Network

This report has been prepared from the threat intelligence gathered by our honeypot network. This honeypot network is today operational in 72 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 12 million attacks a day is being registered across this network of individual honeypots. These attacks are studied, analyzed, categorized, and marked according to a threat rank index, a priority assessment framework that we have developed within Sectrio. 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.