SECTRIO

MALWARE REPORT

Zmutzy: Stealer

Date: 26/04/2021

Meghraj Nandanwar

Overview

Zmutzy is a spyware and information stealer Trojan written in Microsoft's .NET language. Zmutzy is used to spy on the victims by collecting credentials and other information from the infected system. This malware can steal saved user login credentials of Chrome, Opera, Brave, Chromium, Firefox, Outlook, FileZilla, Thunderbird, etc. and look for Crypto Wallets such as Zcash, Armory, Bytecoin, Exodus, Ethereum, etc. Zmutzy is highly obfuscated and uses many anti-analysis techniques to hide itself from being detected.

ANALYSIS

- **Anti-Debugging**

The executable can detect if it is being debugged using IsDebuggerPresent API (Figure.1) and if it found it is being debugged then it will terminate the process. Zmutzy uses sleep to trick automated analysis tools (Figure.2).

```
// Token: 0x06000005 RID: 5
[DllImport("kernel32.dll", EntryPoint = "IsDebuggerPresent")]
private static extern bool \u206F\u206A\u206D\u206D\u206A\u206F\u202A\u200B\u202D\u202A\u200D\u200E\u200B\u202C\u200D\u206E\u206C
  \u206E\u206D\u206A\u202B\u206A\u202C\u200E\u200D\u202C\u202C\u200C\u206B\u202A\u206F\u206B\u202B\u206C\u202D\u200C\u206A\u202E
  \u200D\u202D\u202E();
```

Figure 1: Anti Debug

```
// Token: 0x0600001A RID: 26 RVA: 0x000D1CE8 File Offset: 0x000CFEE8
static void \u206F\u206C\u206D\u202E\u202B\u206C\u202E\u206E\u202E\u202B\u200E\u206C\u202B\u200C\u206F\u202C\u202D\u202E\u200D
  \u202C\u200E\u206D\u206E\u202B\u200B\u200D\u202C\u200C\u206F\u200E\u202B\u200B\u202A\u206B\u202D\u202A\u202A\u206E\u206C
  \u206D\u202E(int A_0)
{
    Thread.Sleep(A_0);
}

// Token: 0x0600001B RID: 27 RVA: 0x000D1CFC File Offset: 0x000CFEFC
static bool \u202A\u202D\u202C\u206E\u202A\u206C\u200D\u200D\u200D\u206C\u202D\u206D\u206E\u206A\u206F\u200C\u202A\u202D\u206A
  \u202B\u206B\u202C\u206A\u202B\u206F\u206E\u202E\u200B\u200E\u200E\u202C\u202A\u206E\u200B\u200E\u200D\u202D\u200D\u202E
  \u206B\u202E()
{
    return Debugger.IsAttached;
}

// Token: 0x0600001C RID: 28 RVA: 0x000D1D10 File Offset: 0x000CFF10
static bool \u206F\u200E\u206B\u206D\u202D\u202E\u200D\u202C\u202D\u202E\u206D\u206A\u200C\u206B\u202B\u206F\u200C\u206C\u200B
  \u200B\u200B\u200E\u206D\u206F\u206F\u200E\u206D\u202D\u200F\u206A\u200D\u200C\u206F\u200D\u200F\u206B\u200C\u206C\u200B
  \u206E\u202E()
{
    return Debugger.IsLogging();
}

// Token: 0x0600001D RID: 29 RVA: 0x000D1C48 File Offset: 0x000CFE48
static void \u202C\u206D\u202C\u202A\u206E\u202E\u206C\u200C\u206E\u200B\u202A\u200D\u200C\u200C\u206F\u200C\u200D\u202E\u206D
  \u202E\u206B\u206E\u202D\u206C\u206C\u206A\u202E\u200C\u206A\u202A\u206B\u206E\u200D\u202B\u206A\u206E\u200B\u200F\u202C
  \u202C\u202E(string A_0)
{
    Environment.FailFast(A_0);
}
```

Figure 2: Debugger Detected

- **Anti-VM and Anti-SB**

Zmutzy can detect Virtual Machines and Sandbox environments to bypass the automated analysis tools. It is uses WMI (Windows Management Instrumentation) and registry to get information about the system to detect Virtual Machines. For anti-sandbox it is checks system username with common username used by sandboxes.

```
// Token: 0x060000CB RID: 203 RVA: 0x000D87C8 File Offset: 0x000D69C8
public static bool AntiVM()
{
    bool flag3;
    ManagementScope managementScope;
    for (;;)
    {
        IL_01:
        uint num = 1429374157U;
        for (;;)
        {
            uint num2;
            switch ((num2 = (num ^ 409109254U)) % 65U)
            {
            case 0U:
                num = (num2 * 1102360987U ^ 3323642899U);
                continue;
            case 1U:
            {
                bool flag = !b希g成cM顾T太孙司.\u202E\u206B\u202A\u200F\u206C\u206B\u202B\u200B\u200F\u206A\u200D\u206D\u206B\u206C\u206D\u202A\u202E\u206D
                  \u202D\u200C\u206F\u202C\u202B\u206C\u202D\u206A\u202C\u200C\u206A\u202A\u200D\u202B\u202C\u202B\u206B\u202E\u206A\u200B\u202E\u206B\u202E(b
                  希g成cM顾T太孙司.\u206D\u202E\u200C\u202E\u206F\u200C\u202D\u200C\u200B\u200F\u200B\u202E\u206D\u200C\u202D\u202E\u206F\u200E\u200D\u206B
                  \u206D\u200B\u202B\u202E\u206F\u206B\u206B\u202E\u202B\u206D\u202D\u202A\u206C\u200B\u206D\u202B\u202D\u202D\u202C\u202D\u202E(b希g成cM顾T太孙
                  司.regGet(T首官Vn氏首gENu孙F.\u202E\u206B\u200E\u200C\u202E\u202D\u200E\u206F\u200E\u200C\u206D\u206A\u202D\u202C\u206B\u202E\u206C\u206A
                  \u200D\u206D\u202D\u202D\u200B\u200C\u202C\u200F\u200D\u200B\u206F\u206A\u200D\u200D\u202A\u202C\u206D\u202B\u202A\u202D\u200F\u206F
                  \u202E<string>(1209282009U), T首官Vn氏首gENu孙F.\u202D\u200D\u200D\u202E\u202B\u200E\u202B\u206E\u200C\u202C\u202A\u200B\u206E\u206D\u200D
                  \u200D\u206E\u206C\u202C\u206F\u206B\u202B\u202C\u206D\u200C\u200E\u200F\u206F\u200F\u202A\u200C\u202A\u202A\u206E\u206B\u200E\u206D\u202B
                  \u206D\u206A\u202E<string>(2730411548U))), T首官Vn氏首gENu孙F.\u206C\u202A\u200C\u200D\u200F\u206D\u200D\u206E\u200E\u200E\u202A\u206F\u206D
                  \u200B\u200E\u206E\u202E\u200D\u200E\u200D\u206F\u200C\u206F\u202C\u200D\u206C\u206B\u200E\u206C\u202C\u200B\u200C\u200D\u200E\u206E\u206B
                  \u206B\u202B\u200C\u202E<string>(1802230542U));
                num = 864791635U;
                continue;
```

Figure 3: Obfuscated Anti-VM function

```
// Token: 0x060000CA RID: 202 RVA: 0x000D80A0 File Offset: 0x000D62A0
public static bool AntiSB(string 家SB庶商Tw是p)
{
    bool result;
    for (;;)
    {
        IL_01:
        uint num = 745892869U;
        for (;;)
        {
            uint num2;
            bool flag2;
            bool flag9;
            switch ((num2 = (num ^ 1757107515U)) % 60U)
            {
            case 0U:
            {
                bool flag = b希g成cM顾T太孙司.\u202E\u206B\u202A\u200F\u206C\u206B\u202B\u200B\u200F\u206A\u200D\u206D\u206B\u206C\u206D\u202A\u202E\u206D\u202D
                  \u200C\u206F\u202C\u202B\u206C\u202D\u206A\u202C\u200C\u206A\u202A\u200D\u202B\u202C\u202B\u206B\u202E\u206A\u200B\u202E\u206B\u202E(b希g成cM
                  顾T太孙司.\u206D\u202E\u200C\u202E\u206F\u200C\u202D\u200C\u200B\u200F\u200B\u202E\u206D\u200C\u202D\u202E\u206F\u200E\u200D\u206B\u206D\u200B
                  \u202B\u202E\u206F\u206B\u206B\u202E\u202B\u206D\u202D\u202A\u206C\u200B\u206D\u202B\u202D\u202D\u202C\u202D\u202E(家SB庶商Tw是p), T首官Vn氏首
                  gENu孙F.\u206B\u206F\u200E\u206C\u202B\u202B\u202A\u200B\u206A\u206A\u206B\u202E\u200C\u202E\u206F\u206E\u206D\u200D\u200F\u202B\u202B\u200E
                  \u202E\u206C\u202E\u200F\u200C\u200F\u202B\u206D\u200D\u202D\u202D\u206F\u200B\u206D\u206B\u200E\u200F\u202E<string>(1006575898U));
                num = 1399527310U;
                continue;
            }
```

```
            int num3 = 50;
            StringBuilder stringBuilder;
            b希g成cM顾T太孙司.GetUserName(stringBuilder, ref num3);
            num = (num2 * 432899480U ^ 1787928874U);
            continue;
        }
```

Figure 4: Obfuscated Anti-SB function

- **Injection Flow**

Packed sample unpacks itself and if it is unable to find any analysis tricks, it injects malicious payload inside the AppLaunch.exe (Figure.5).

Process	Description	Image Path	Life Time	Company	Owner	Command
abc.exe (10744)	Grim Dawn Stash Manager	C:\Users\Meghraj\...		420WeedWizard Delivery	DESKTOP-E53ES...	"C:\Users\Meghraj\Desktop\abc.exe"
abc.exe (9480)	Grim Dawn Stash Manager	C:\Users\Meghraj\...		420WeedWizard Delivery	DESKTOP-E53ES...	"C:\Users\Meghraj\Desktop\abc.exe"
AppLaunch.exe (1388)	Microsoft .NET ClickOnce Launch Utility	C:\Windows\Micro...		Microsoft Corporation	DESKTOP-E53ES...	"C:\Windows\Microsoft.NET\Framework\v4.0.30319\AppLaunch.exe"
WerFault.exe (6776)	Windows Problem Reporting	C:\Windows\SysW...		Microsoft Corporation	DESKTOP-E53ES...	C:\Windows\SysWOW64\WerFault.exe -u -p 1388 -s 76
InstallUtil.exe (3972)	.NET Framework installation utility	C:\Windows\Micro...		Microsoft Corporation	DESKTOP-E53ES...	"C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil.exe"
InstallUtil.exe (1056)	.NET Framework installation utility	C:\Windows\Micro...		Microsoft Corporation	DESKTOP-E53ES...	"C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil.exe"
WerFault.exe (8540)	Windows Problem Reporting	C:\Windows\SysW...		Microsoft Corporation	DESKTOP-E53ES...	C:\Windows\SysWOW64\WerFault.exe -u -p 3972 -s 872
AppLaunch.exe (11160)	Microsoft .NET ClickOnce Launch Utility	C:\Windows\Micro...		Microsoft Corporation	DESKTOP-E53ES...	"C:\Windows\Microsoft.NET\Framework\v4.0.30319\AppLaunch.exe"
InstallUtil.exe (10520)	.NET Framework installation utility	C:\Windows\Micro...		Microsoft Corporation	DESKTOP-E53ES...	"C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil.exe"
WerFault.exe (1132)	Windows Problem Reporting	C:\Windows\SysW...		Microsoft Corporation	DESKTOP-E53ES...	C:\Windows\SysWOW64\WerFault.exe -u -p 10520 -s 80
AppLaunch.exe (9752)	Microsoft .NET ClickOnce Launch Utility	C:\Windows\Micro...		Microsoft Corporation	DESKTOP-E53ES...	"C:\Windows\Microsoft.NET\Framework\v4.0.30319\AppLaunch.exe"
WerFault.exe (5808)	Windows Problem Reporting	C:\Windows\SysW...		Microsoft Corporation	DESKTOP-E53ES...	C:\Windows\SysWOW64\WerFault.exe -u -p 9752 -s 80
InstallUtil.exe (6796)	.NET Framework installation utility	C:\Windows\Micro...		Microsoft Corporation	DESKTOP-E53ES...	"C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil.exe"
InstallUtil.exe (4188)	.NET Framework installation utility	C:\Windows\Micro...		Microsoft Corporation	DESKTOP-E53ES...	"C:\Windows\Microsoft.NET\Framework\v4.0.30319\InstallUtil.exe"
WerFault.exe (6112)	Windows Problem Reporting	C:\Windows\SysW...		Microsoft Corporation	DESKTOP-E53ES...	C:\Windows\SysWOW64\WerFault.exe -u -p 6796 -s 876
AppLaunch.exe (10124)	Microsoft .NET ClickOnce Launch Utility	C:\Windows\Micro...		Microsoft Corporation	DESKTOP-E53ES...	"C:\Windows\Microsoft.NET\Framework\v4.0.30319\AppLaunch.exe"

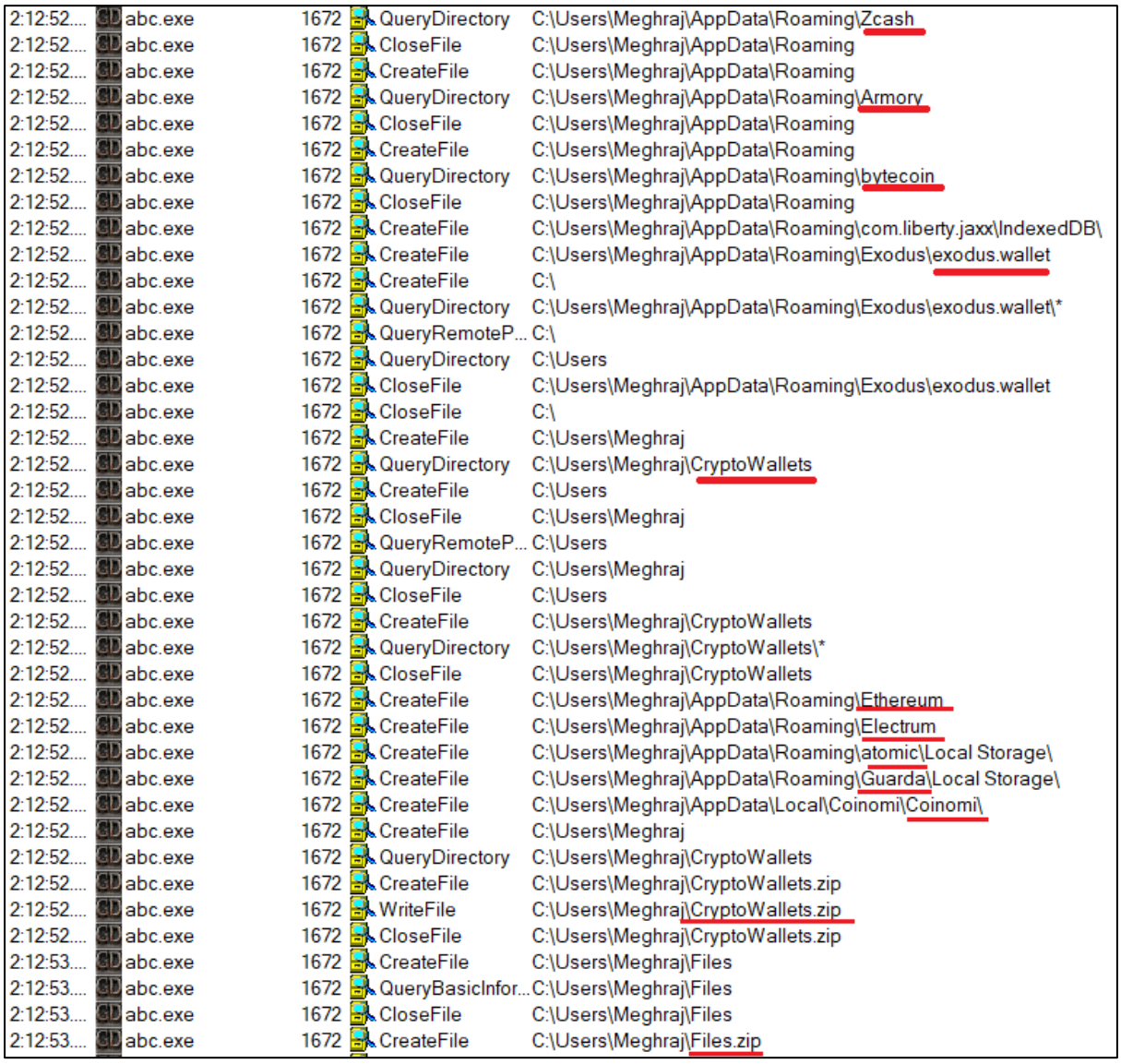
Figure 5: Injection Flow

```
// Token: 0x02000017 RID: 23
// (Invoke) Token: 0x06000309 RID: 777
private delegate int DelegateResumeThread(IntPtr handle);

// Token: 0x02000018 RID: 24
// (Invoke) Token: 0x0600030D RID: 781
private delegate bool DelegateWow64SetThreadContext(IntPtr thread, int[] context);

// Token: 0x02000019 RID: 25
// (Invoke) Token: 0x06000311 RID: 785
private delegate bool DelegateSetThreadContext(IntPtr thread, int[] context);

// Token: 0x0200001A RID: 26
// (Invoke) Token: 0x06000315 RID: 789
private delegate bool DelegateWow64GetThreadContext(IntPtr thread, int[] context);

// Token: 0x0200001B RID: 27
// (Invoke) Token: 0x06000319 RID: 793
private delegate bool DelegateGetThreadContext(IntPtr thread, int[] context);

// Token: 0x0200001C RID: 28
// (Invoke) Token: 0x0600031D RID: 797
private delegate int DelegateVirtualAllocEx(IntPtr handle, int address, int length, int type, int protect);

// Token: 0x0200001D RID: 29
// (Invoke) Token: 0x06000321 RID: 801
private delegate bool DelegateWriteProcessMemory(IntPtr process, int baseAddress, byte[] buffer, int bufferSize, ref int bytesWritten);

// Token: 0x0200001E RID: 30
// (Invoke) Token: 0x06000325 RID: 805
private delegate bool DelegateReadProcessMemory(IntPtr process, int baseAddress, ref int buffer, int bufferSize, ref int bytesRead);

// Token: 0x0200001F RID: 31
// (Invoke) Token: 0x06000329 RID: 809
private delegate int DelegateZwUnmapViewOfSection(IntPtr process, int baseAddress);

// Token: 0x02000020 RID: 32
// (Invoke) Token: 0x0600032D RID: 813
private delegate bool DelegateCreateProcessA(string applicationName, string commandLine, IntPtr processAttributes, IntPtr threadAttributes, bool
  inheritHandles, uint creationFlags, IntPtr environment, string currentDirectory, ref a泽太COQ行顾i的.StartupInformation startupInfo, ref a泽太COQ行顾i
  的.ProcessInformation processInformation);
```

Figure 6: APIs used for Process Injection

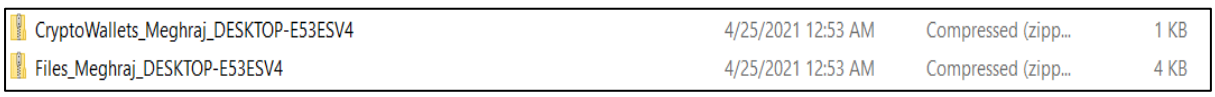
- **Cryptocurrency Wallet Stealing**

Unpacked sample search Crypto Wallets (Figure.7) and all the files in the user desktop and collect those files and save them in zip folder (Figure.8).



2:12:52....	abc.exe	1672	QueryDirectory	C:\Users\Meghraj\AppData\Roaming\Zcash
2:12:52....	abc.exe	1672	CloseFile	C:\Users\Meghraj\AppData\Roaming
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj\AppData\Roaming
2:12:52....	abc.exe	1672	QueryDirectory	C:\Users\Meghraj\AppData\Roaming\Armory
2:12:52....	abc.exe	1672	CloseFile	C:\Users\Meghraj\AppData\Roaming
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj\AppData\Roaming
2:12:52....	abc.exe	1672	QueryDirectory	C:\Users\Meghraj\AppData\Roaming\bytecoin
2:12:52....	abc.exe	1672	CloseFile	C:\Users\Meghraj\AppData\Roaming
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj\AppData\Roaming\com.liberty.jaxx\IndexedDB\
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj\AppData\Roaming\Exodus\exodus.wallet
2:12:52....	abc.exe	1672	CreateFile	C:\
2:12:52....	abc.exe	1672	QueryDirectory	C:\Users\Meghraj\AppData\Roaming\Exodus\exodus.wallet*
2:12:52....	abc.exe	1672	QueryRemoteP...	C:\
2:12:52....	abc.exe	1672	QueryDirectory	C:\Users
2:12:52....	abc.exe	1672	CloseFile	C:\Users\Meghraj\AppData\Roaming\Exodus\exodus.wallet
2:12:52....	abc.exe	1672	CloseFile	C:\
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj
2:12:52....	abc.exe	1672	QueryDirectory	C:\Users\Meghraj\CryptoWallets
2:12:52....	abc.exe	1672	CreateFile	C:\Users
2:12:52....	abc.exe	1672	CloseFile	C:\Users\Meghraj
2:12:52....	abc.exe	1672	QueryRemoteP...	C:\Users
2:12:52....	abc.exe	1672	QueryDirectory	C:\Users\Meghraj
2:12:52....	abc.exe	1672	CloseFile	C:\Users
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj\CryptoWallets
2:12:52....	abc.exe	1672	QueryDirectory	C:\Users\Meghraj\CryptoWallets*
2:12:52....	abc.exe	1672	CloseFile	C:\Users\Meghraj\CryptoWallets
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj\AppData\Roaming\Ethereum
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj\AppData\Roaming\Electrum
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj\AppData\Roaming\atomic\Local Storage\
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj\AppData\Roaming\Guarda\Local Storage\
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj\AppData\Local\Coinomi\Coinomi\
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj
2:12:52....	abc.exe	1672	QueryDirectory	C:\Users\Meghraj\CryptoWallets
2:12:52....	abc.exe	1672	CreateFile	C:\Users\Meghraj\CryptoWallets.zip
2:12:52....	abc.exe	1672	WriteFile	C:\Users\Meghraj\CryptoWallets.zip
2:12:52....	abc.exe	1672	CloseFile	C:\Users\Meghraj\CryptoWallets.zip
2:12:53....	abc.exe	1672	CreateFile	C:\Users\Meghraj\Files
2:12:53....	abc.exe	1672	QueryBasicInfor...	C:\Users\Meghraj\Files
2:12:53....	abc.exe	1672	CloseFile	C:\Users\Meghraj\Files
2:12:53....	abc.exe	1672	CreateFile	C:\Users\Meghraj\Files.zip

Figure 7: Checks for Crypto Wallets



CryptoWallets_Meghraj_DESKTOP-E53ESV4	4/25/2021 12:53 AM	Compressed (zipp...	1 KB
Files_Meghraj_DESKTOP-E53ESV4	4/25/2021 12:53 AM	Compressed (zipp...	4 KB

Figure 8: Saved Harvested Information

- **Credential Stealing**

After injecting payload into AppLaunch.exe it looks for user credentials from web browser, mail application and various sources from which attacker can obtain user credentials (Figure.9).

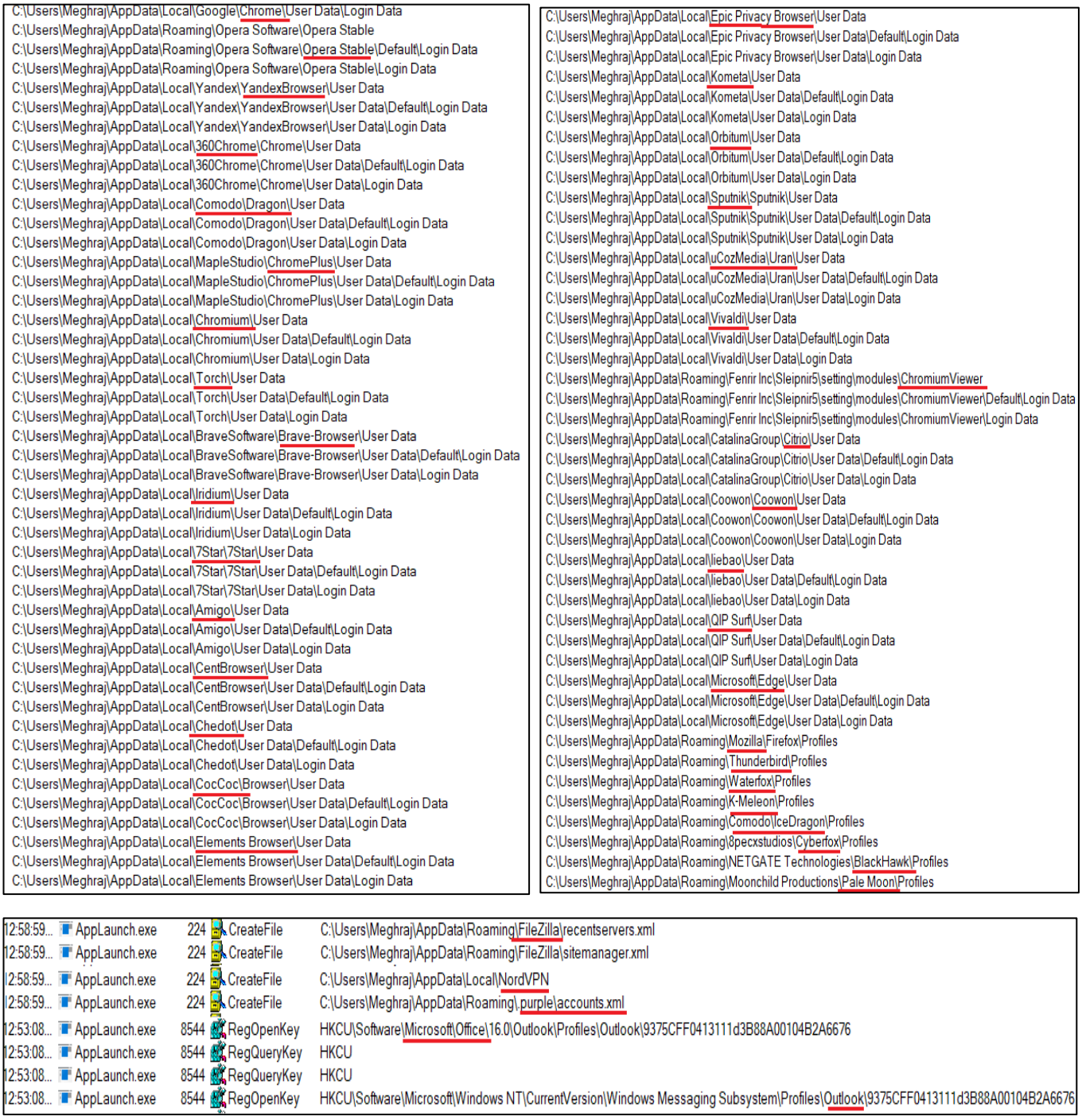


Figure 9: Searching for User Credentials

Network Activity

AppLaunch.exe connects to the icanhazip.com (Figure.10) to check the victim's IP and there is only little network activity take place by sample.

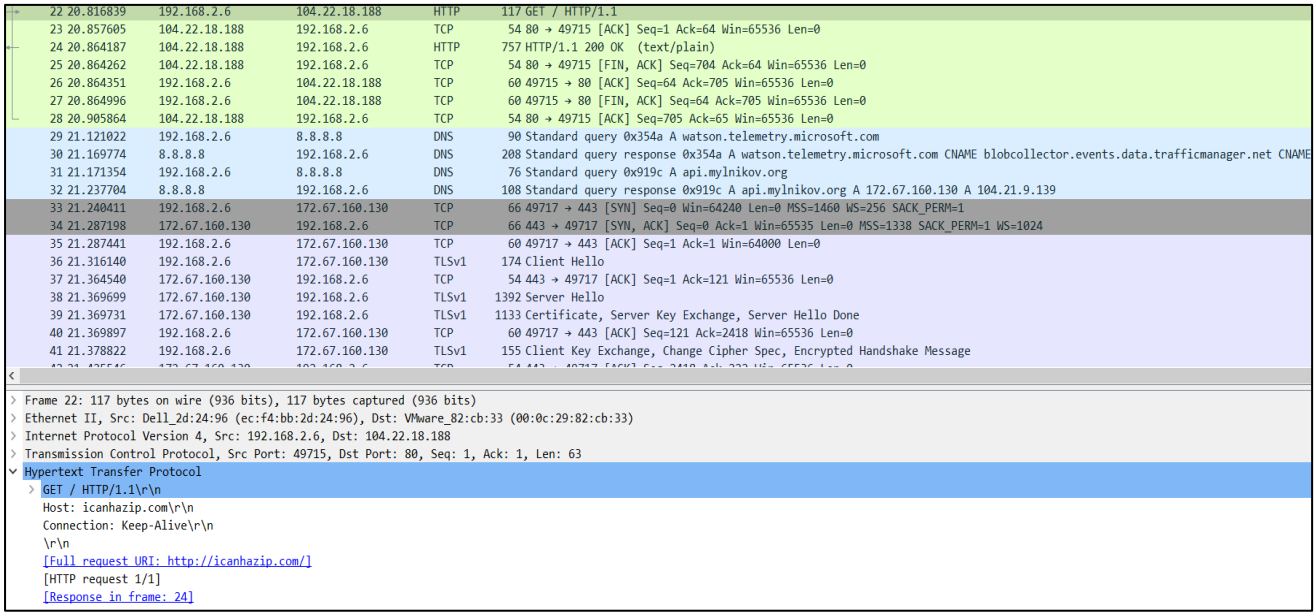


Figure 10: Network Activity

The contacted IP addresses include:

172.67.160.130	104.22.18.188
104.21.9.139	104.22.19.188

Sample Details

File	Hash Value (MD5)
Packed Sample	CA59FFD2BE3593BD0DCE304A72482575

MITRE ATTACK TECHNIQUES

TACTIC	ID	NAME
Execution	T1047	Windows Management Instrumentation
Defense Evasion	T1055	Process Injection
Defense Evasion	T1027	Obfuscated Files or Information
Defense Evasion	T1027.002	Software Packing
Defense Evasion	T1497	Virtualization/Sandbox Evasion
Defense Evasion	T1036	Masquerading
Defense Evasion	T1140	Deobfuscate /Decode Files or Information
Defense Evasion	T1562.001	Disable or Modify Tools
Credential Access	T1003	OS Credential Dumping
Credential Access	T1552.002	Credentials in Registry
Credential Access	T1552.001	Credentials in Files
Discovery	T1082	System Information Discovery

Discovery	T1083	File and Directory Discovery
Discovery	T1518.001	Security Software Discovery
Discovery	T1016	System Network Configuration Discovery
Collection	T1005	Data from Local System
Collection	T1114	Email Collection
Collection	T1560	Archive Collected Data
Command and Control	T1105	Ingress Tool Transfer
Command and Control	T1573	Encrypted Channel

Sectrio Protection

Sectrio detects the Spyware-Trojan malware 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 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.