LA MENACE FANTÔME

Nicolas Brulez – nicolas.brulez@kaspersky.fr
Senior Malware Researcher – Global Research and Analysis Team Kaspersky Lab France

mots-clés : CODES MALICIEUX / REVERSE ENGINEERING / SHELLCODE / EXPLOIT / TROJANS / MEMORY / STEALTH

Au début du mois de mars, nous avons reçu un rapport d'un chercheur indépendant sur des infections en masse d'ordinateurs d'entreprises, suite à la visite de sites (légitimes) d'actualités russes. Les symptômes étaient les mêmes dans chaque cas : apparition de requêtes vers des sites tiers et création de fichiers incompréhensibles (probablement chiffrés) sur les machines.
Le mécanisme d'infection utilisé par ce logiciel malveillant s'est avéré très difficile à identifier. Les sites utilisés pour propager l'infection étaient hébergés sur différentes plates-formes et avaient des architectures différentes. L'utilisation d'une faille commune était alors écartée. De plus, il était impossible de reproduire les infections. Toutefois, nous avons trouvé un point commun à tous ces sites d'actualités.

1 Vecteur d'infection

Pour continuer l'analyse, nous avons choisi deux sites supposés distribuer la menace, puis nous les avons aspirés (**wget** *is your friend*) régulièrement afin de les inspecter en *offline*. Nous n'avons découvert aucune trace d'injection de code : ni Javascript tiers, ni iframe et aucune redirection caractéristique d'une infection. Le seul point commun était légitime : l'utilisation des services de publicité de la société AdFox.

Après avoir creusé la piste de la bannière de publicité, nous avons découvert que le *malware* était chargé via les bandes-annonces d'AdFox. Les sites n'étaient pas compromis, mais leurs bannières l'étaient. Chaque bande-annonce contient un Javascript, mais certains étaient modifiés et contenaient une *iframe* qui redirigait vers un site malveillant contenant un exploit Java. Vous pouvez voir sur la figure 2 l'injection d'une iframe.

L'analyse du fichier JAR (exploit) a permis d'identifier l'utilisation d'une vulnérabilité Java (CVE-2011-3544) utilisée depuis novembre par les cybercriminels pour cibler les utilisateurs Windows et Mac OS. Les *exploits* pour cette faille font partie des plus efficaces à l'heure actuelle. On les retrouve donc dans les « ExploitPacks » les plus populaires.

```
<div id="first-rc-bb" class="column">
  <a id="mm" name="mm"></a>
<div class="banner_14">
  <noindex>
    <script type="text/javascript">
      1
      2  <!--
      3  if (typeof(pr) == 'undefined') { var pr = Math.floor(Mat
      4  if (typeof(document.referrer) != 'undefined') {
      5    if (typeof(afReferrer) == 'undefined') {
      6      afReferrer = escape(document.referrer);
      7    }
      8  } else {
      9    afReferrer = '';
     10  }
     11  var addate = new Date();
     12  var dl = escape(document.location);
     13  var pr1 = Math.floor(Math.random() * 1000000);
     14
     15  document.write('<div id="AdFox_banner_'+pr1+'"><\/div>')
     16  document.write('<div style="visibility:hidden; position:
     17
     18  AdFox_Scroll(pr1,'http://ads.adfox.ru/147017/prepareCode
     19  // -->
    </script>
    <div id="AdFox_banner_496559"></div>
    <div style="visibility:hidden;
    position:absolute;">
```

Figure 1 : Site utilisant le service d'Adfox

Cependant, il ne s'agissait pas d'un copier-coller habituel, mais d'une version unique de l'exploit que nous n'avions encore jamais vue. De plus, la charge finale (*payload*) était aussi unique comme nous allons le voir maintenant.

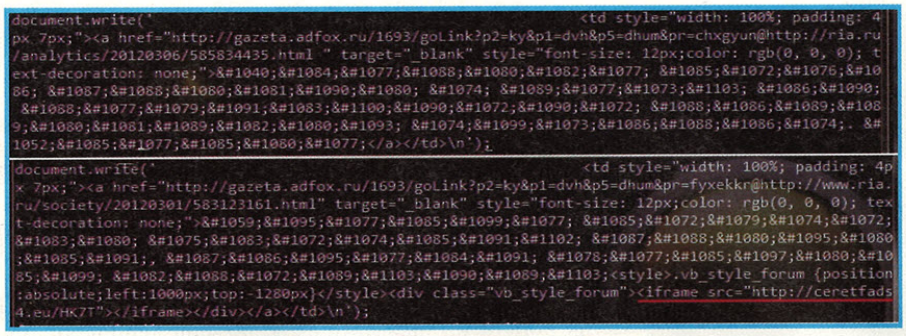

Figure 2 : Iframe injectée

2 La menace fantôme : tout en mémoire

En règle générale, le fonctionnement d'un tel exploit consiste à sauvegarder un fichier malveillant, généralement un *dropper* ou *downloader* sur le disque dur. Ce nouveau fichier, une fois exécuté par l'exploit, s'occupe de télécharger le malware final sur la machine : un *bot*, un cheval de Troie, ou autre faux antivirus. Toutefois, dans notre cas, aucun fichier n'est apparu sur le disque de la machine infectée. Aucun fichier n'était créé, pourtant les symptômes étaient bien présents.

Contrairement aux exploits classiques, le shellcode télécharge, injecte et exécute une dll cryptée directement dans le processus **javaw.exe** sans jamais en faire une copie sur le disque. Il est donc impossible de trouver la menace une fois la machine éteinte. L'URL de la dll malveillante à télécharger est présente sous forme chiffrée dans l'iframe ajoutée au Javascript de la bannière AdFox compromise :

```
<applet code="Applet.class" archive="/0GLMFss"> </applet>
```

Après avoir réussi à injecter et exécuter la dll (présente seulement en mémoire) le processus javaw.exe commence alors à envoyer des requêtes qui ressemblent à des recherches Google : « ? Recherche hl = fr & source = hp & q =% s & aq = f & aqi = & aql = & oq = »...

Ces requêtes incluent des données sur l'historique de navigation prises à partir du navigateur de l'utilisateur, ainsi que de nombreuses informations techniques sur le système infecté. Le malware ici présent, tout comme CodeRed et Slammer, n'utilise pas de fichier sur le disque de la machine infectée et réside en mémoire seulement. Ce type de malware ne survit pas au redemarrage de la machine puisqu'il n'existe aucune copie. Cela peut rendre l'analyse « post mortem » délicate.

De plus, cela n'est pas un handicap pour ce malware, puisqu'en fonction des réponses du serveur de contrôle, un autre malware est téléchargé puis exécuté : le cheval de Troie bancaire « Lurk ».

Il est important de noter que la décision d'installer ce Trojan est prise côté serveur, et non localement.

De plus, il est fort probable que la victime retourne sur le site d'actualités une fois la machine redemarrée, se traduisant par une nouvelle infection tout en mémoire. N'utilisant aucun fichier sur le disque, il devient beaucoup plus difficile de détecter la menace « fantôme ». En cas de non-détection de l'exploit, le bot est virtuellement invisible pour l'utilisateur.

3 Cheval de Troie bancaire : Lurk

Le malware Trojan-Spy.Win32.Lurk peut être installé sur une machine de deux manières différentes. Soit en utilisant les commandes **regsrv32** et **netsh add helper dll**, soit via **ShellconOverlayIdentifiers** dans le registre système. Lurk installe ensuite ses propres modules sous forme de dll chiffrées.

02FD0000	00002000				Map	R	R	
02FE0000	00004000				Map	RW	RW	\Device\HarddiskVolume1\Documents and Settings\
02FF0000	00008000				Map	RW	RW	\Device\HarddiskVolume1\Documents and Settings\
03000000	00003000				Priv	RW Gua:	RW	
0304B000	00001000				Priv	RW Gua:	RW	
0304C000	00004000			stack of th:	Priv	RW Gua:	RW	
03050000	00010000				Map	RW	RW	
03090000	00003000				Priv	RW	RW	
03290000	00017000				Priv	RWE	RWE	
032FE000	00001000				Priv	RW Gua:	RW	
032FF000	00001000			stack of th:	Priv	RW Gua:	RW	
03300000	00001000				Priv	RW	RW	
0331E000	00001000				Priv	RW Gua:	RW	
0331F000	00001000			stack of th:	Priv	RW Gua:	RW	
03350000	00001000				Map	R	R	
03380000	00002000				Map	R	R	
038B0000	00002000				Map	R	R	
22990000	00550000				Priv	RW	RW	
27EE0000	00AB0000				Priv	RW	RW	
32990000	00C00000				Priv	RW	RW	
36990000	00530000				Map	R	R	\Device\HarddiskVolume1\PROGRA~1\Java\jre6\bin\
37390000	006A0000				Map	RW	RW	\Device\HarddiskVolume1\PROGRA~1\Java\jre6\bin\
37F90000	000D0000				Map	RW	RW	\Device\HarddiskVolume1\PROGRA~1\Java\jre6\bin\
38390000	00010000				Priv	R E	RW	

Figure 3 : Processus JAVAW.exe après injection

```
aVw_dll         db 'vw.dll',0
                db   0
aWapi_dll       db 'wapi.dll',0
                db   0
                db   0
                db   0
aSetup_dll      db 'setup.dll',0
                db   0
                db   0
aEnv_dll        db 'env.dll',0
aP10_dll        db 'p10.dll',0
aTheme_dll      db 'theme.dll',0
                db   0
                db   0
aHttp_dll       db 'http.dll',0
                db   0
                db   0
                db   0
aNet_dll        db 'net.dll',0
aMm_dll         db 'mm.dll',0
                db   0
aPool_drv       db 'pool.drv',0
                db   0
                db   0
                db   0
aSta_dll        db 'sta.dll',0
aCore_dll       db 'core.dll',0
                db   0
                db   0
                db   0
aMi_dll         db 'mi.dll',0
                db   0
aDlg_dll        db 'dlg.dll',0
```

Figure 4 : Liste de modules supplémentaires pour Lurk

L'analyse des modules supplémentaires utilisés par Lurk nous permet de comprendre la finalité de l'attaque : le vol de données sensibles des utilisateurs pour accéder aux services en ligne de plusieurs grandes banques russes.

Kaspersky Lab a détecté la premiere version de ce malware en juillet 2011. Sur la base de notre analyse du protocole utilisé par Lurk pour communiquer avec les serveurs de contrôle, nous avons pu déterminer que sur une période de plusieurs mois, ces serveurs ont traité les requêtes d'environ 300 000 machines infectées.

4 AdFox : le maillon faible

Après avoir terminé l'analyse technique, nous avons contacté Adfox pour les informer de l'incident. Ils ont rapidement réagi pour détecter et nettoyer l'injection des bannières publicitaires.

Lors des investigations, il fut déterminé que les cybercriminels avaient utilisé le compte d'un client Adfox pour modifier le code des bannières en ajoutant une iframe permettant la redirection des visiteurs vers un site malveillant. Seul ce client (compromis) diffusait du contenu dangereux.

Une fois le code modifié, ils ont pu attaquer tous les visiteurs des sites qui affichaient les bannières infectées.

Après la modification du code dans l'une des bannières, ils étaient en mesure d'attaquer non seulement les utilisateurs du site d'actualités, mais aussi les visiteurs d'autres sites affichant cette même bannière. En conséquence, des dizaines de milliers d'utilisateurs ont pu être attaqués. Les bannières des autres clients Adfox ne furent quant à elles non modifiées, et ne représentaient aucune menace pour leurs visiteurs.

Conclusion

Il s'agit d'une attaque unique. Les cybercriminels ont utilisé leur propre code de téléchargement et d'injection de dll dans un exploit, sans écrire de fichier sur le disque lors de l'exploitation et infection de la machine. La dll fonctionne entièrement dans le processus de confiance Java.

L'utilisation d'un réseau publicitaire (compromis) est une méthode très efficace pour cibler un grand nombre d'utilisateurs. En effet, de nombreux sites légitimes deviennent alors des vecteurs d'infection.

Bien que l'attaque ciblait les utilisateurs russes, rien n'empeche les criminels de l'adapter aux autres pays à l'aide de réseaux publicitaires locaux compromis pour assurer la grande distribution du code malveillant. La charge finale, Lurk dans notre exemple, pourrait aussi être remplacée très simplement.

Pour se protéger contre cette menace, nous suggérons fortement aux utilisateurs d'installer un patch qui corrige la vulnérabilité Java CVE-2011-3544. C'est actuellement le seul moyen fiable pour éviter l'infection.

Comme nous l'avons mentionné ci-dessus, les exploits pour CVE-2011-3544 sont les plus efficaces et peuvent être utilisés pour installer une grande variété de programmes malveillants.

Les techniques évoluent pour assurer la furtivité et l'infection du plus grand nombre, toujours dans une optique lucrative. ■