

```
[jdaniel@archbox ~]$ whoami
```



Entre chinos, fraudes y hashes: Integridad en sistemas críticos



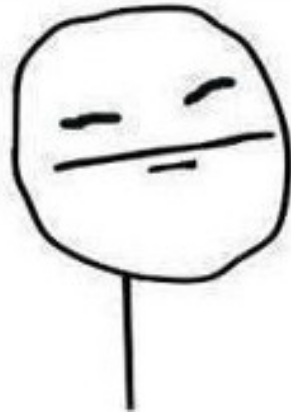


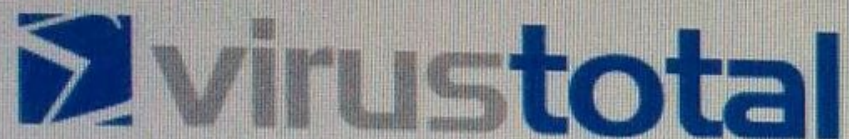
- **Permiso para dar esta charla**
- **No se revela informacion sensible**
- **Los demos no representan las arquitecturas reales.**

Motivación

3 Historias

1.- “El chino”





SHA256: 51f8bdf240eaa4a909ca74abdb45a7d9794ef83daf8eeeb9adc3543dbe3aa16a

File name: linux_2_66_26_elx5paets_6_64elhsveo_linux_XMP_ROOT_sfedree_66x

Detection ratio: 24 / 55

Analysis date: 2014-11-11 11:06:46 UTC (7 months, 2 weeks ago)

[Analysis](#)[File detail](#)[Additional information](#)[Comments](#)

1

[Votes](#)

Antivirus

Result

Ad-Aware

Linux.DDOS.Flood.A

AhnLab-V3

Linux/Backdoor.1223123.B

Avast

ELF:Elknot-AS [Trj]

BitDefender

Linux.DDOS.Flood.A

CAT-QuickHeal

Linux.DnsAmp.a586

ELF Linux/BillGates



Apache Struts2

CVE-2013-2251

“introduces the possibility to inject server side code.”





2.- “Token mágico”



===== .jsp ===== .jpg ===== .jpg ===== .jpg ===== .jpg
===== .jpg ===== .jpg ===== .jpg ===== .jpg



JSPSpy

 122.119.120.18:8180/downloads/JspSpy.jsp





122 点击可后退，按住可查看历史记录 | [copy](#)

[Logout](#) | [File Manager](#) | [DataBase Manager](#) | [Execute Command](#) | [Shell OnLine](#) | [Back Connect](#) | [Java Reflect](#) | [Eval Java Code](#) | [Port Scan](#) | [Download Remote File](#) | [Clipboard](#) | [Port Map](#) | [Others](#) | [JSP Env](#)

File Manager - Current disk "/" total (unknow)

Current Directory

[Web Root](#) | [Shell Directory](#) | [New Directory](#) | [New File](#) | [Disk \(/\)](#)

Name	Last Modified	Size	Read/Write/Execute
 Goto Parent			
 META-INF	2014-03-27 03:10:57	--	true / true / unknow
 WEB-INF	2014-03-27 03:10:58	--	true / true / unknow
 images	2014-03-27 03:10:57	--	true / true / unknow
<input type="checkbox"/> getAgentAndroid.html	2014-03-27 03:10:56	64B	true / true / unknow
<input type="checkbox"/> getAndroid.html	2014-03-27 03:10:56	61B	true / true / unknow
<input type="checkbox"/> index.html	2014-03-27 03:10:56	1.12K	true / true / unknow
<input type="checkbox"/> index.jsp	2014-03-27 03:10:56	106B	true / true / unknow
<input type="checkbox"/> msky-10001000.apk	2014-07-29 05:01:10	5.99M	true / true / unknow
<input type="checkbox"/> openurl.html	2014-03-27 03:10:56	3.98K	true / true / unknow
<input type="checkbox"/> style.css	2014-03-27 03:10:56	843B	true / true / unknow
<input type="checkbox"/> umetrip.apk	2014-07-29 05:01:10	5.99M	true / true / unknow
<input type="checkbox"/> umetripcwt.apk	2014-03-27 03:10:57	4.87M	true / true / unknow

[Pack Selected](#) - [Delete Selected](#)

<https://github.com/tennc/webshell/blob/master/jsp/JspSpy.jsp>

Consecuencias

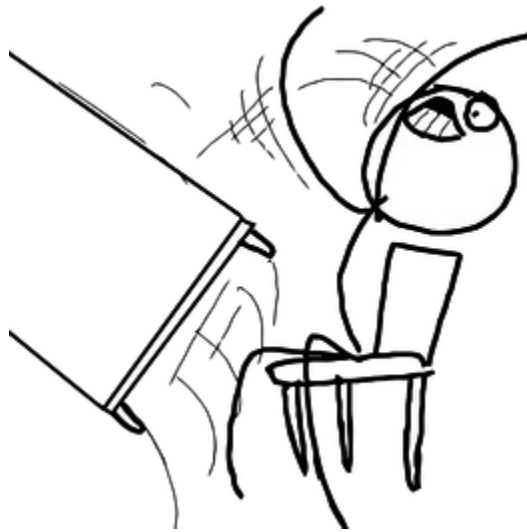


**I KNOW YOU HAVEN'T
DONE ANYTHING WRONG**

**BUT YOU SHOULD LOOK DIRECTLY
AT ME SO WE CAN SUSPECT YOU**

quickmeme.com

3.- “A pesar de ..”



WEBSHELLS WEBSHELLS

EN TODOS LADOS

How Russian Hackers Spiked the Currency Exchange Rate

Don Reisinger

Feb 08, 2016



Russian hackers found a way to dramatically alter a currency exchange rate—in just 14 minutes.

Launching a virus known as Corkow Trojan against Russia-based Energobank, a group of Russian hackers altered the value of the ruble against the dollar, *Bloomberg* is [reporting](#), citing an interview with Group-IB, the company that investigated the attack. The virus, which hit Energobank in Feb. 2015, allowed the hackers to buy more than \$500 million "at non-market rates," according to the report. The move was enough for the ruble's exchange rate to jump from 55 to 66 rubles per dollar before it settled back down.



SwiftIn4KUltraHD @SwiftOnSecurity · 9 jun.

What percentage of real-world companies could effectively neutralize a malevolent trusted administrator with Windows/*NIX/VMware skillsets?

6% 1%

7% 0.1%

24% 0.01%

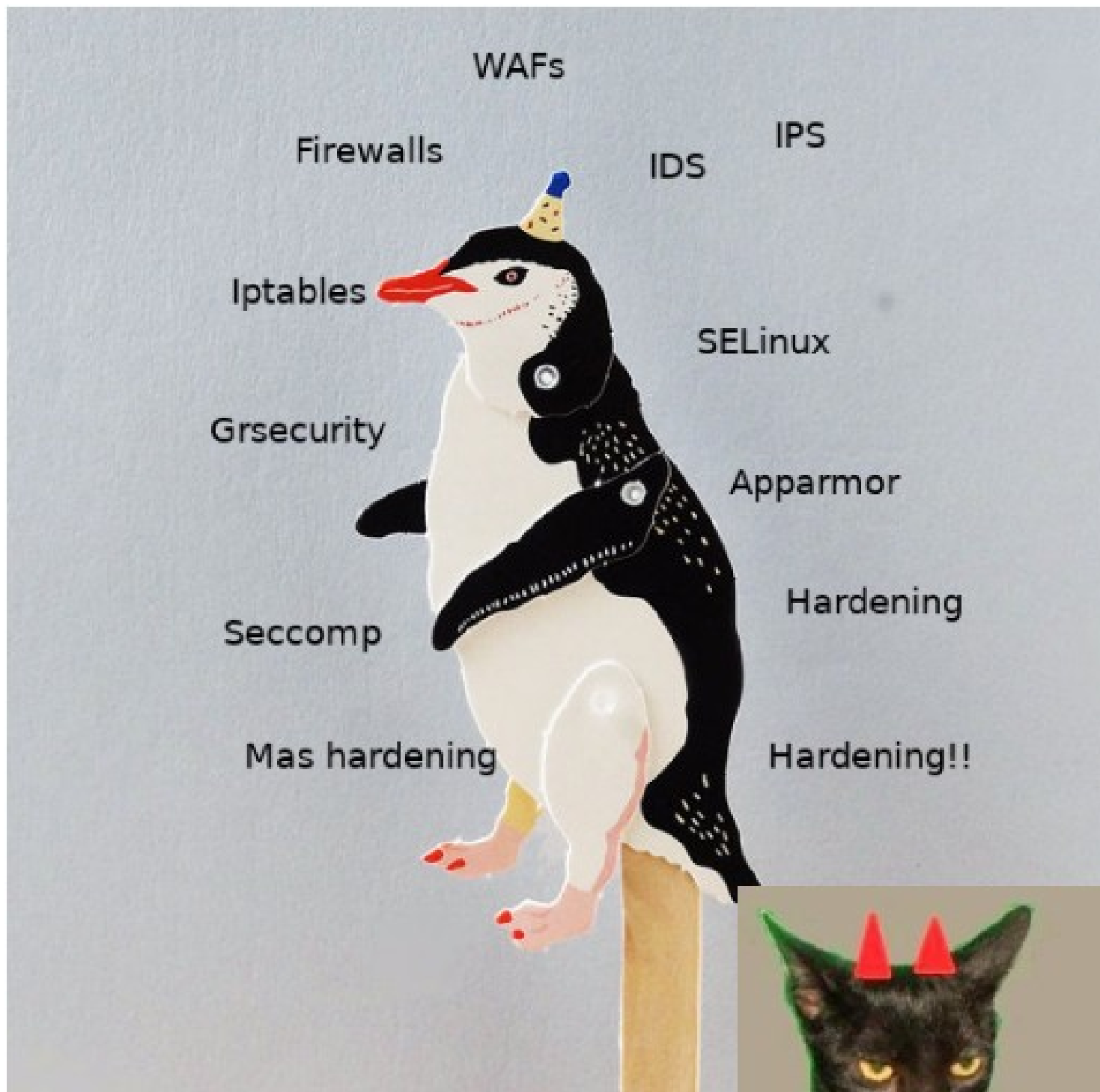
63% Maybe a few banks

2,747 votos • Resultados finales

← 42

↻ 34

♥ 60



Entonces

- Upload de archivos
- Ejecucion de archivos
- Modificacion de archivos confiables
- Atacantes Internos o ex-empleados

Hablemos de Confianza



Desde Kernel version 2.6.30 (Junio 2009)



```
[*] Integrity Measurement Architecture(IMA)
    Default template (ima-ng (default)) --->
    Default integrity hash algorithm (SHA1 (default)) --->
[ ] Enable multiple writes to the IMA policy (NEW)
[ ] Enable reading back the current IMA policy (NEW)
[*] Appraise integrity measurements
[*] Require all keys on the .ima keyring be signed (deprecated)
[ ] Create IMA machine owner blacklist keyrings (EXPERIMENTAL) (NEW)
[ ] Load X509 certificate onto the '.ima' trusted keyring (NEW)
[*] EVM support
```

IMA

(Integrity Measurement Architecture)

Hooks en el Kernel para medir la integridad `sys_exec`,
`sys_open`, `sys_mmap`

- Verifica integridad de archivos antes de ser accedidos **(Collect)**
- Almacena los valores en listas internas del Kernel **(Store)**
- Hash en un log que puede ser consultado **(Attest)**

cat políticas > /sys/kernel/security/ima/policy

Políticas :

measure func=FILE_MMAP mask=MAY_EXEC

measure func=BPRM_CHECK mask=MAY_EXEC

measure func=FILE_CHECK mask=MAY_READ

```
[root@prueba ~]# cat /sys/kernel/security/ima/ascii_runtime_measurements
```

```
10 bad737ce9d611e8374a471ccaf3947b51963b8af ima-ng sha1:4d868fb1fa8f8be10d42627f062c90fdf9d2cec5 boot_aggregate
10 23975b10c53e626b34254afd71c861f5d9570c69 ima-ng sha1:10a3bb3f0de17a773b0612cce90ec8f1938442af /usr/bin/bash
10 3974c1e62b5dee0d276658dc4f4cadd556bc8760 ima-ng sha1:4c69859c62daec5e8e7978aa1f8334b6003edcd3 /usr/lib64/ld-2.17.so
10 6a9953d143918686bff31b00ef4017e548f6c723 ima-ng sha1:29d2074abc75c8a5e1cf1361c4a6c9c583a99b88 /usr/lib64/libtinfo.so.5.9
10 409393a5a757759ba3ad8c503d551b7d5569f17d ima-ng sha1:d611ddd0729ea9422c4b9bdbb2fa8a63fba897e6 /usr/lib64/libdl-2.17.so
10 f2258901ecb204abf0f0d26f1ca6e9a9f715cad6 ima-ng sha1:efc455a07a54bfd1bd158d2c6063ff9eca5f6c4 /usr/lib64/libc-2.17.so
10 a2e8b429b8eee5a2e36d05d3e60f08fbfb1c8b04 ima-ng sha1:3a07f5d53785c7207f1d53657d65bbf86fb1686f /usr/lib64/libnss_files-2.17.so
10 e3eb1f76fcf148d3416c6a8df46c2d4ed59c5fb6 ima-ng sha1:f3362e50370756cdcbf0532407fd0d868724e3f3 /usr/bin/id
10 4289af40ded01a99e803579b4d0a6fbc67140e55 ima-ng sha1:79f50f805b118af664597a554d51219df6f1c461 /usr/lib64/libselinux.so.1
10 c511b44f72045cb62545f92a77f9d5915fff5b30 ima-ng sha1:91c7e62f3e16691e53c49f045fab4be2f9cfe4ce /usr/lib64/libpcre.so.1.2.0
10 abbc13ef2a738ac8df1c33f1baa5c233a67371aa ima-ng sha1:8e4373315c6af96f8e9cda711e8cea40a758b039 /usr/lib64/libpthread-2.17.so
10 e86e62ac7a712125c7e222c805e4ab2346d7a84c ima-ng sha1:5b2cb2adea6d8448e58a4c1393c421796dbe9038 /usr/bin/hostname
10 bf8497999e8c9c7784bd12f82fe8b414bc61b5ac ima-ng sha1:55cd99d0a7d663cae07e7fadbe84b2ba0411d6fe /usr/lib64/libnsl-2.17.so
10 abccc345b46e11129277b24bcf05e5b71a013834 ima-ng sha1:357aec665c5fd77179efea474ac025bb4d425baf /usr/bin/grep
10 5202f0c81e61abdd62f4d2d92c477f7884b697c6 ima-ng sha1:1100c00ae088221a100f0d21af88daf52db3cd25 /usr/bin/tty
10 864d099e3f90608ad5eda5e73d276588f6c66f2e ima-ng sha1:c5209b6992b5ca6a8b152ddd739b2fcf5e0dcdd /usr/bin/tput
10 1a70f8dbec6b97229023083bc354d709ae854d7a ima-ng sha1:c7f87ae681e30a203f89a71dd521fd3709ec85f7 /usr/bin/dircolors
10 77780231132a699c14e28cce9404a54ccee4cfaf ima-ng sha1:9120ffedc3290dd81f80ccffcb8693ce16d76c37 /usr/sbin/consoletype
```

Process Control Register| El hash del hash del archivo | Subsystema usado (ima)| hash del archivo

Remote attestation



IMA-appraisal

- Valida el registro local guardado en el atributo extendido **security.ima** de cada archivo (Appraise)
- Niega el acceso al archivo si el valor no concuerda
- Comandos propios de IMA se usan para agregar hash o firmas a los archivos

<https://sourceforge.net/p/linux-ima/ima-evm-utils/ci/master/tree/>

Políticas:

appraise func=FILE_MMAP mask=MAY_EXEC

uid=1000

appraise func=BPRM_CHECK mask=MAY_EXEC

uid=1000

```
[root@prueba ~]# su - tomcat
```

```
Last login: Wed May 24 10:11:41 EDT 2017 on pts/0
```

```
[tomcat@prueba ~]$ ls
```

```
-bash: /bin/ls: Permission denied
```

```
[root@prueba tomcat]# evmctl sign --imahash /bin/ls
```

```
[root@prueba tomcat]# getfattr -h -e hex -d -m security /bin/ls
```

```
# file: .bash_logout
```

```
security.ima=0x0166296908ae73bd0b72b71fb15b413e9c7b81c69d
```

```
[root@prueba ~]# su - tomcat
```

```
[tomcat@prueba ~]$ ls
```

```
aa.bas  a.sh  s.sh  ss.py
```


¿Que necesito?

- Tener activado IMA en el kernel
- Habilitar IMA en el grub
- Montar los FS con la opcion de i_version
- Enteder las politicas de IMA
https://www.kernel.org/doc/Documentation/ABI/testing/ima_policy
- Agregar las politicas de iMA al initramfs
- Tiempo , luchar con pobre y casi nula documentacion



Volviendo a las 3 Historias...

DEMO CHINO

DEMO TOKEN MAGICO

BONUS “OFFLINE ATTACK”

BONUS 2 IMA EN HONEYPOT

Conclusiones

- Resulta ser una solución efectiva si conoces tu entorno
- No es infalible, ataques a IMA:
[https://seclab.stanford.edu/pcl/cs259/projects/cs259_final_lavina_jayesh/C S259_report_lavina_jayesh.pdf](https://seclab.stanford.edu/pcl/cs259/projects/cs259_final_lavina_jayesh/C%20S259_report_lavina_jayesh.pdf)
- Complementar con TPM,EVM
- Capas inferiores aun son una "caja negra"

No confies en nadie, si no tienes control total de tu infraestructura

