



REPORTE DE AUDITORÍA DE SEGURIDAD INFORMÁTICA PARA COPA - REVIEW

Noviembre 30, 2018

PROPIEDADES DEL DOCUMENTO

Título	Black Box Penetration Testing Report - Review
Versión	V 2.0
Autor	Daniel Ortiz
Pen-testers	Daniel Ortiz
Revisado por	
Clasificación	Confidencial

CONTROL DE VERSIÓN

Versión	V 2.0
Fecha	30/11/2018
Autor	Confidencial
Descripción	Review

ÍNDICE DE CONTENIDO

REPORTE DE AUDITORÍA DE SEGURIDAD INFORMÁTICA PARA COPA - REVIEW	0
1 Reporte ejecutivo	5
1.1 Alcance de la auditoría	5
1.2 Objetivos de la auditoría	5
1.3 Suposiciones	6
1.4 Cronograma de la auditoría	6
1.5 Resumen de hallazgos	6
2. Metodología	7
2.1 Planificación	7
2.2 Explotación	8
2.3 Reporte	8
3. Narrativa del ataque	8
3.1 Scanning and fingerprinting from the web	9
3.2 Juicy information from shodan	9
3.3 Detalles de sistemas encontrados en shodan	11
3.4 Available endpoints	12
3.5 Información interesante en código fuente	15
3.6 Posible oracle application server disponible	17
4. Hallazgos en servidores	19

ÍNDICE DE TABLAS, IMÁGENES Y GRÁFICOS

Tabla 1. Cronograma de la auditoría.	6
Tabla 2. Resumen de hallazgos.	6
Imagen 1. Penetration testing methodology.	7
Tabla 3. Scanning and fingerprinting from the web.	8
Imagen 2. Script para consultar API de SHODAN.	9
Imagen 3. Resultado del script SHODAN.	9
Imagen 4. Posible diagrama de red de beta-checkin.	10
Imagen 5. Endpoints en ./src/api/checkIn.js.	11
Imagen 6. Código de respuesta del request.	11
Imagen 7. Respuesta al consultar /parameters/ffprograms.	11
Imagen 8. Respuesta luego del request.	12
Imagen 9. IP hard code en código fuente.	12
Imagen 10. Nmap scan en IP 18.209.36.128.	13
Imagen 11. Brute Force attack en puerto RDP.	14
Imagen 12. Import de script en Metasploit Framework.	14
Imagen 13. Configuración de opciones.	15
Imagen 14. Hallazgos luego de la ejecución del scanner.	15

Gráfico 1.1. Hallazgos en Servidores.

Tabla 4.1.1 API Token Authentication improper configuration.

Tabla 4.1.2 IP Address available on source code with RDP exposed

Tabla 4.1.3 Possible OAS Default SOAP Configuration

1 REPORTE EJECUTIVO

El presente informe se ha realizado con la finalidad de revisar si fueron aplicadas las contramedidas recomendadas en la auditoría realizada a la infraestructura indicada por COPA durante el período 17 al 21 de Septiembre, 2018. En el mismo se determina una mejora significativa en la eliminación de las vulnerabilidades detectadas durante la auditoría realizada en Septiembre, sin embargo un par de nuevas vulnerabilidades han sido identificadas, esto es natural ya que todos los días surgen nuevas vulnerabilidades. Recomendamos incluir a estos sistemas en la lista de sistemas que son revisados semanalmente en el testeo de gestión de vulnerabilidad. Adicionalmente es importante proteger a estos sistemas y recomendamos activar tanto el doble factor de autenticación (servicios MSS-TAS actual) y un servicio que proteja contra actividad sospechosa (como el MSS-EDR de GLESEC). Otras recomendaciones se encuentran más detalladas en el final de este documento.

1.1 ALCANCE DE LA AUDITORÍA

Este análisis de seguridad cubre los recursos suministrados por el cliente vía email luego de confirmar la ejecución de las pruebas en un ambiente beta o pre productivo.

1.2 OBJETIVOS DE LA AUDITORÍA

El presente estudio tiene como finalidad poder identificar aquellos equipos y servicios que se encuentran vulnerables y que pueden ser accesibles desde internet. Las vulnerabilidades fueron clasificadas según el impacto y el nivel de amenaza.

1.3 SUPOSICIONES

Durante la elaboración de este informe se tuvo presente que algunos equipos fueron accesibles desde Internet, esto permitió realizar pruebas más exhaustivas en estos equipos.

1.4 CRONOGRAMA DE LA AUDITORÍA

A continuación se detallan el plan de actividades y su tiempo de duración para la presente auditoría:

Actividad	Fecha de Inicio	Fecha de Finalización
Penetration testing	17/09/2018	25/09/2018
Elaboración del informe	26/09/2018	28/09/2018
Revisión	27/11/2018	30/09/2018

Tabla 1. Cronograma de la auditoría.

1.5 RESUMEN DE HALLAZGOS

Riesgo	Número de vulnerabilidades reportadas	Número de vulnerabilidades corregidas	Número de vulnerabilidades encontradas (nuevas)
--------	---------------------------------------	---------------------------------------	---

Info	78	-	-
Bajo	2	2	0
Mediano	7	7	2
Alto	0	-	0

Tabla 2. Resumen de hallazgos.

Fue posible identificar nuevas vulnerabilidades de rango mediano en 2 servidores.

2. METODOLOGÍA

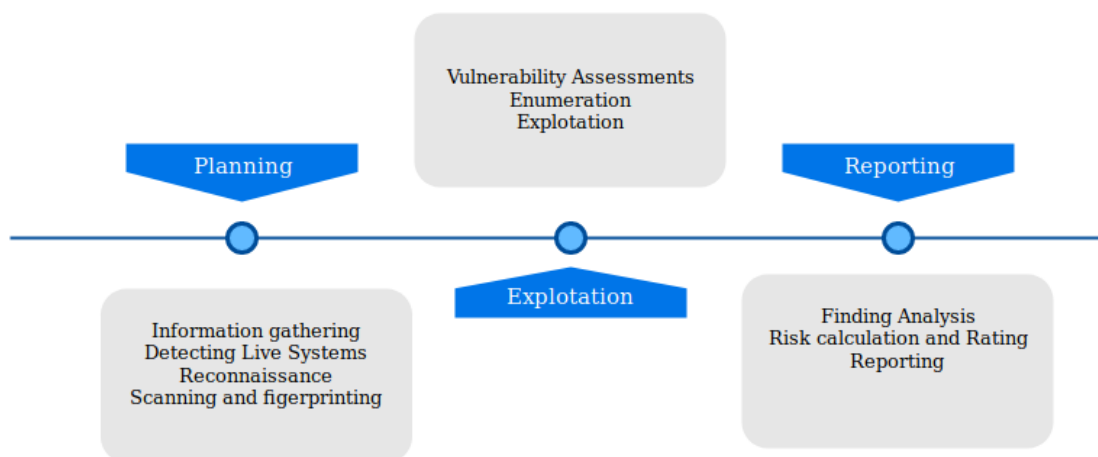


Imagen 1. Penetration testing methodology.

2.1 PLANIFICACIÓN

Durante las pruebas se recolectó información de los equipos disponibles utilizando diferentes fuentes. Adicionalmente aprovechando la conectividad de los equipos de GLESEC desde AWS, se efectuaron técnicas de fingerprinting y scanning con el objetivo de recolectar mayor información.

2.2 EXPLOTACIÓN

Luego de la información recolectada durante la etapa de fingerprinting y scanning, se procedió a buscar la forma de explotar alguna vulnerabilidad en los sistemas auditados.

2.3 REPORTE

Posterior a la obtención de los resultados, se procedió a la medición de riesgo y a la elaboración del presente informe.

3. NARRATIVA DEL ATAQUE

3.1 SCANNING AND FINGERPRINTING FROM THE WEB

A continuación se presentan los equipos que se encuentran visibles desde internet y la información que un atacante podría obtener haciendo un escaneo de puertos.

IP	SYSTEM TYPE	OS INFORMATION	OPEN PORTS			
			PORT	PROTOCOL	SERVICE NAME	STATUS
94.188.209.136	SERVER	EC2	80	TCP	HTTP	OPEN
			443	TCP	HTTPS	OPEN
			80	TCP	HTTP	OPEN

94.188.209.42	SERVER	EC2	443	TCP	HTTPS	OPEN
---------------	--------	-----	-----	-----	-------	------

Tabla 3. Scanning and fingerprinting from the web.

3.2 JUICY INFORMATION FROM SHODAN

En medio de la fase de information gathering del reporte anterior, se pudo encontrar información interesante de los servidores en la base de datos de SHODAN. Utilizando la API y un sencillo script en Python reproducimos nuevamente las pruebas.

```
from providers.search_engine import ShodanEngine
from providers import config

def main():
    """ Main method """
    ips = ["94.188.209.136", "94.188.209.41"]
    for i in ips:
        search_ip(i.__str__())

def search_ip(ip):
    """ Search for specific IP Address """
    client = ShodanEngine(config.API_KEY_SHODAN)
    api = client.init_shodan()
    try:
        host = api.host(ip)
        for item in host['data']:
            print("""
                Port: {}
                Banner: {}

                """.format(item['port'], item['data']))
    except Exception:
        print("No information available about " + ip)

# TODO: [2] Add more methods for shodan

if __name__ == '__main__':
    main()
```

Imagen 2. Script para consultar API de SHODAN.

Luego de la ejecución del script se pudo notar que la respuesta de la API de SHODAN fue mucho menor a la respuesta obtenida la vez anterior, esto demuestra que menos información relacionada a los servidores se encuentra en la base de datos de SHODAN, esto reduce la cantidad de información que puede obtener un atacante de manera indirecta de los servidores evaluados.

A continuación en la siguiente imagen podemos apreciar los resultados obtenidos luego de la ejecución del script.

3.3 DETALLES DE SISTEMAS ENCONTRADOS EN SHODAN

```
Port: 443
Banner: HTTP/1.1 403 Forbidden
Server: CloudFront
Date: Wed, 28 Nov 2018 18:40:52 GMT
Content-Type: text/html
Content-Length: 556
Connection: keep-alive
X-Cache: Error from cloudfront
Via: 1.1 6c9a37f8749807266eb4262a6e987f26.cloudfront.net (CloudFront)
X-Amz-Cf-Id: LVYBRsn8TfAXpn8cJhPxZx0i3EsCCIEESNaK_qSZA3Xy3H0PpoX2Sw==

Port: 80
Banner: HTTP/1.1 302 Found
Location: https://94.188.209.42/
Connection: Keep-Alive
Content-Length: 0
```

Imagen 3. Resultado del script SHODAN

Podemos notar en los resultados obtenidos que la información devuelta por la API no es muy relevante, la única información que revela esta API son los puertos 80 y 443 que fueron identificados con el escaneo de puertos y revela también que los archivos son servidos desde CloudFront AWS. Se podría realizar el siguiente diagrama:

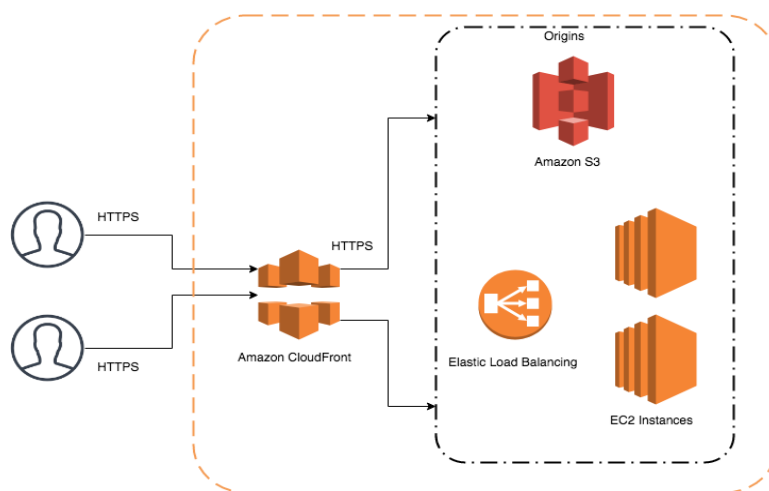


Imagen 4. Posible diagrama de red de beta-checkin.

3.4 AVAILABLE ENDPOINTS

Uno de los puntos a revisar del informe anterior fueron los endpoints de la app, en esta revisión se pudo notar que los endpoints fueron movidos de directorio y los métodos fueron reescritos. En la siguiente imagen podemos notar los endpoints disponibles en el directorio `/src/api/checkIn.js`

```
export default {
  flight: {
    searchFlight: (number, lang) =>
      axios
        .get(`/reservation?pnr=${number}`, {
          headers: {
            'Accept-Language': lang,
          },
        })
        .then(res => res.data),
  },
  reservation: {
    findReservation: ({ code, lastName, lang }) =>
      axios
        .get(`/reservation/${code}/${lastName}`, {
          headers: {
            'Accept-Language': lang,
          },
        })
        .then(res => res.data),
  },
  checkIn: {
    checkInAllPassengers: ({ pnr, data, lang }) =>
      axios
        .post(`/reservation/checkin/${pnr}`, data, {
          headers: {
            'Accept-Language': lang,
          },
        })
        .then(res => res.data),
  },
}
```

Imagen 5. Endpoints en `/src/api/checkIn.js`.

A continuación procedimos a revisar si la API tiene habilitado algún tipo de mecanismo de autenticación, realizamos un GET a la URL <https://api-checkin.copaair.com> dando como respuesta 403 (Forbidden) al request y mostrando en el response un mensaje de token de autenticación tal como se muestra en las siguientes imágenes.

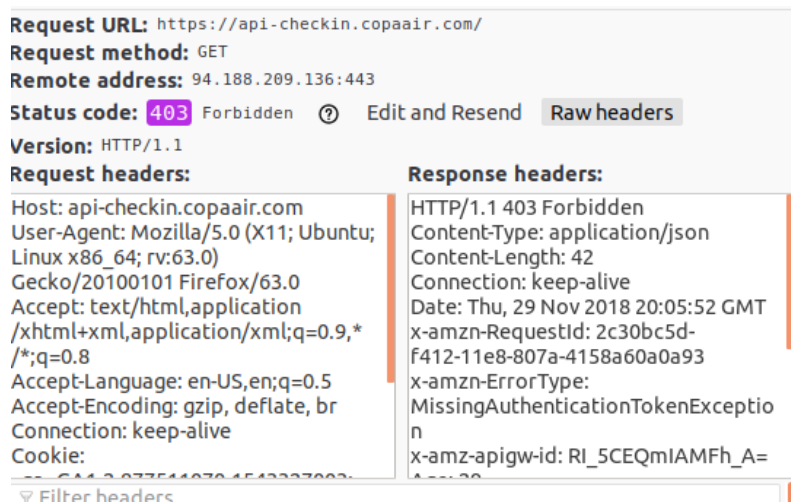


Imagen 6. Respuesta de request.

En esta imagen se puede notar el mensaje de missing authentication token el cual indica que esta sección de la aplicación necesita autenticación.

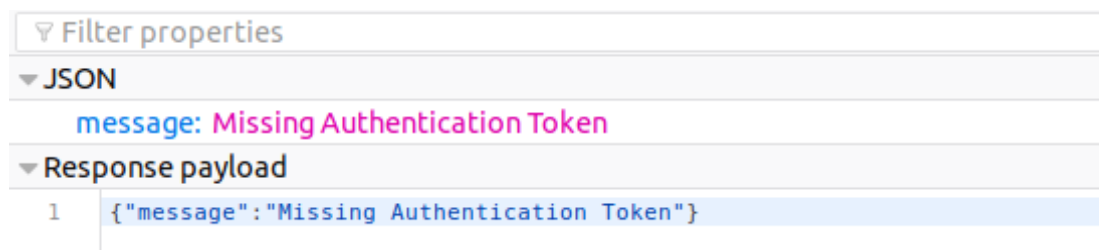


Imagen 7. Mensaje de Missing Token.

A continuación realizamos la misma prueba pero con otro endpoint en este caso **/parameters/ffprograms**, para este caso en específico arroja 200 (Ok).

```
Request URL: https://api-checkin.copaaair.com/parameters/ffprograms
Request method: GET
Remote address: 94.188.209.136:443
Status code: 200 OK ⓘ Edit and Resend Raw headers
Version: HTTP/1.1
```

Imagen 8. Código de estado al consultar /parameters/ffprograms.

Como podemos notar en la imagen también nos arroja resultados, por la información contenida en la respuesta parece estar relacionada a los tipos de membresías.

```
▼ Filter properties
▼ JSON
  ▼ 0: {...}
    frequentFlyerCode: CM
    frequentFlyerDesc: ConnectMiles
    usedByCopa: Y
  ▼ 1: {...}
    frequentFlyerCode: UA
    frequentFlyerDesc: Mileage Plus
    usedByCopa: Y
  ▼ 2: {...}
    frequentFlyerCode: AC
    frequentFlyerDesc: Aeroplan
    usedByCopa: Y
  ▼ 3: {...}
```

Imagen 9. Resultados luego del request

Es curioso notar que en la sección principal de la app solicite autenticación y en las secciones subyacentes no pase lo mismo, recomendamos revisar con el proveedor de desarrollo si este comportamiento es el esperado.

3.5 INFORMACIÓN INTERESANTE EN CÓDIGO FUENTE

También fue posible identificar en una sección del código fuente de la aplicación, una dirección IP de un servicio HTTPS.

```
A = Object(u.createStore)(g.a, Object(f.composeWithDevTools)(Object(u.applyMiddleware)(S, d.a)));  
...S.run(E.a), Object(w.a)(A), Object(_a)(A), Object(b.a)("https://18.209.36.128"),  
i.a.render(a.a.createElement(s.c, {  
.....history: 0.a
```

Imagen 10. IP hard code en código fuente.

A continuación procedimos a realizar un escaneo de puertos sobre la dirección IP en cuestión y logramos obtener información interesante de los servicios expuestos, fue posible identificar el servicio RDP.

A continuación podemos notar los servicios expuestos en el servidor y el resultado devuelto por el escáner.


```

1 Host is up, . received user-set. (0.0017s latency).
2 Scanned at 2018-11-27 12:23:49 UTC. for: 7s
3 Not shown: 999 filtered ports
4 Reason: 999 no responses
5 PORT: . . . . . STATE SERVICE . . . . . REASON
6 3389/tcp open . ms-wbt-server syn-ack
7 |.ssl-cert: Subject: commonName=EC2AMAZ-VI9NT13
8 |.Issuer: commonName=EC2AMAZ-VI9NT13
9 |.Public Key type: rsa
10|.Public Key bits: 2048
11|.Signature Algorithm: sha256WithRSAEncryption
12|.Not valid before: 2018-10-31T18:01:36
13|.Not valid after: 2019-05-02T18:01:36
14|.MD5: . . . 9bff.17a8.01bc.7e16.ef79.9432.5401.d9b1
15|.SHA-1: . a0c6.9170.1b68.1f09.2176.8359.8b41.6821.fda6.4961
16|.-----BEGIN CERTIFICATE-----
17|.MIIC4jCCAcgAwIBAgIQe04PJMaPFqhJHonXjYxVwzANBgkqhkiG9w0BAQsFADAa
18|.MRgwFgYDVQ0DEw9FQzJBTUFAVZJOU5UMTMwHhcNMTgxMDMxMTgwMTM2WncNMTkw
19|.NTAyMTgwMTM2WjAaMRgwFgYDVQ0DEw9FQzJBTUFAVZJOU5UMTMwggEiMA0GCSCqG
20|.S1b3DQEBQUAA4IBDwAwggEKAoIBAQCvm3ja50u1SEcXsmhHNq6HV0hyJ2XhPxBU
21|. /Enoo1XguaqG4pV16AAJrBRhCTwmBAE6GUyP4oI8vkC2hMy7Wo9Tf2PzYBLHaGvH
22|. oKcZeS0PEYHsqfAY19XBvd+UZTJXh24IUxsZ0k5/YQqa/2fA7AML7rxp3F+4Qom
23|. 02Kw5l1CUpDV4MLKJf0/84iKUL3q9U8Zq3LK6BySR0AtK/EBcHVWzFOA+aUxI3gN
24|. 0s+I4Vd8LfgIJZ5sT04IalkhL+DPZ/fwBEPeSeatdpEMqB0Xh2pSaJCOFZMej848
25|. 0Ax22a4/F3acJRbsyVg1rvqxz8oVms2ZcCb7xYTqGBysIGuUOHPjAgMBAAgJDA1
26|. MBMGA1UdJQQMMAoGCCsGAQUFBwMBMAsGA1UdDwQEAwIEMDANBgkqhkiG9w0BAQsF
27|. AAOCAQ0EAClyBtZKZJeomYOLIpGy6+TPEBudUptVA6gKYF94pk0LDYNR0WaKaTFAL
28|. Mq+ypM0ExptX8qXa32p3xNIBk5lsSkYwepwkFR1UTOMNynBYgl0XSYg0EucEYsX8
29|. km6ghvX2ETXj5dPGDnb9mnzs9qm3AJdskLM5+aRAC94hZFSAR+h/Ufm+z6owGlnI
30|. ioTM1gl7gZr8gmW/sF0Z8P3mH8f6CIFGPXmE6IvmSP57UombGXPsqNZoYPNBaqwL
31|. ct2wLKLhWYyTJfuEg/duzJhUhPEYvzTsBBafMH7tNcFXISKlZ26C+LH7/Nuq3lAF
32|. vhsc53oAGTzqNCFCLi0Xl5Rcn9diJQ==
33|.-----END CERTIFICATE-----
34|.ssl-date: 2018-11-27T12:23:56+00:00; .0s from scanner time.
35
36 Host script results:
37 |_clock-skew: .mean: .0s, .deviation: .0s, .median: .0s
38

```

Imagen 10. Nmap scan en IP 18.209.36.128.

Procedimos a realizar un ataque de fuerza bruta en el servicio RDP. El ataque duró alrededor de 35 horas y fueron probadas 425.542 contraseñas, en ningún momento durante el ataque fue reseteada la conexión o no se identificó alguna acción para mitigar el ataque. En la siguiente imagen podemos notar parte de los intentos realizados.

```
hydra (http://www.thc.org/thc-hydra) starting at 2018-11-27 12:37:18
[WARNING] the rdp module is currently reported to be unreliable, most likely against new Windows version. Please test, report - and if possible, fix.
[DATA] max 1 task per 1 server, overall 1 task, 14344399 login tries (l1:p:14344399), ~14344399 tries per task
[DATA] attacking rdp://18.209.36.128:3389/
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "12345" - 2 of 14344400 [child 0] (0/1)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "123456789" - 3 of 14344401 [child 0] (0/2)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "password" - 4 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "loveyou" - 5 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "princess" - 6 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "1234567" - 7 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "rockyou" - 8 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "12345678" - 9 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "abc123" - 10 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "nicole" - 11 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "daniel" - 12 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "babygirl" - 13 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "monkey" - 14 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "lovely" - 15 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "jessica" - 16 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "654321" - 17 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "michael" - 18 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "ashley" - 19 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "qwerty" - 20 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "111111" - 21 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "loveu" - 22 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "000000" - 23 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "michelle" - 24 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "tiger" - 25 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "sunshine" - 26 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "chocolate" - 27 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "password1" - 28 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "soccer" - 29 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "anthony" - 30 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "friends" - 31 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "butterfly" - 32 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "purple" - 33 of 14344402 [child 0] (0/3)
[ATTEMPT] target 18.209.36.128 - login "Administrator" - pass "angel" - 34 of 14344402 [child 0] (0/3)
```

Imagen 11. Brute Force attack en puerto RDP.

Recomendamos revisar si la IP encontrada en el código fuente de la aplicación corresponde con algún servicio y también validar si es necesario la exposición del servicio RDP.

3.6 POSIBLE ORACLE APPLICATION SERVER DISPONIBLE

Parte de la información obtenida en la etapa de Information Gathering y Fingerprinting corresponde a un posible servicio OAS (Oracle Application Server) corriendo en el servidor **beta-checkin**. Encontramos un script en Ruby (escrito por **carnal0nage**) que modificamos para este escaneo y procedimos a importarlo en el framework Metasploit.

```
ec2-user@kali: /usr/share/metasploit-framework/tools/dev$ ./msftidy.rb -f .msf4/modules/auxiliary/admin/oas/cgi_scan_hashmethod.rb
/home/ec2-user/.msf4/modules/auxiliary/admin/oas/cgi_scan_hashmethod.rb - [WARNING] Explicitly requiring/loading msf/core is not necessary
/home/ec2-user/.msf4/modules/auxiliary/admin/oas/cgi_scan_hashmethod.rb - [WARNING] Explicitly using self.class in register_* is not necessary
/home/ec2-user/.msf4/modules/auxiliary/admin/oas/cgi_scan_hashmethod.rb - [WARNING] VERBOSE option is already part of advanced settings, no need to manually
/home/ec2-user/.msf4/modules/auxiliary/admin/oas/cgi_scan_hashmethod.rb - [WARNING] Please use 'MetasploitModule' as the class name (you used
```

Imagen 12. Import de script en Metasploit Framework.

Luego de importar el módulo y configurar cada una de las opciones necesarias para el escaneo procedimos a ejecutarlo.

```
msf auxiliary(oas_cgi_scan_hashmethod) > show info

Name: Oracle Application Server Check
Module: auxiliary/admin/oas_cgi_scan_hashmethod
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
MC <mc@metasploit.com>
CG <cg@carnal0wnage.com>

Basic options:
  Name      Current Setting  Required  Description
  ----      -
DEMOS      false            yes       Enable checks for all the demo pages
Proxies     no               no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS     94.188.209.42    yes       The target address range or CIDR identifier
RPORT      443              yes       The target port (TCP)
SSL         true             no        Negotiate SSL/TLS for outgoing connections
THREADS     1                yes       The number of concurrent threads
VERBOSE     true             yes       Show all error codes or just 200/302
VHOST      beta-checkin.copaair.com no         HTTP server virtual host

Description:
This module scans for common files on an Oracle Application Server
and Oracle Database Server. If you are having issues. set VERBOSE to
TRUE to see all error codes.
```

Imagen 13. Configuración de opciones.

Luego de una serie de intentos fue posible identificar que el servidor responde con código 200 (Ok) a las siguientes URL's.

```
[+] Received 404 for /stax
[+] Found: /soap/ --> Vuln: Oracle SOAP Servlet
[+] Found: /soap/servlet/soaprouter --> Vuln: Oracle SOAP Servlet
[+] Found: /soap/servlet/Spy --> Vuln: Oracle SOAP Servlet
[+] Found: /soap/admin --> Vuln: Oracle SOAP Servlet
[+] Found: /soap/admin/servlet/soaprouter --> Vuln: Oracle SOAP Servlet
[+] Found: /soap/admin/servicemanager --> Vuln: Oracle SOAP Servlet
[+] Found: /soap/admin/providermanager --> Vuln: Oracle SOAP Servlet
```

Imagen 14. Hallazgos luego de la ejecución del scanner.

Recomendamos validar la existencia de este servicio de lado del backend y cambiar la configuración del protocolo SOAP que viene por defecto.

4. HALLAZGOS EN SERVIDORES

A continuación se mencionan las vulnerabilidades encontradas en los servidores.

4.1.1 API TOKEN AUTHENTICATION IMPROPER CONFIGURATION

Nivel	MEDIO
Descripción	Varios endpoints dentro de la aplicación no solicita autenticación vía token y pueden ser consumidos sin ningún tipo de mecanismo de validación.
Impacto	Un atacante podría consultar información de estos endpoints sin ningún tipo de autenticación.
Remediación	Habilitar Token Authentication en los endpoints restantes del sitio.

Tabla 5.1. API Token Authentication improper configuration.

4.1.2 IP ADDRESS AVAILABLE ON SOURCE CODE WITH RDP EXPOSED

Nivel	MEDIO
Descripción	El código fuente de la aplicación contiene una IP con un servicio RDP expuesto que no tiene mecanismos que mitiguen ataques de fuerza bruta.
Impacto	Un atacante podría realizar un ataque de fuerza bruta y en caso de obtener un usuario y password valido del sistema comprometer la seguridad del servidor.
Remediación	Deshabilitar RDP o limitar el acceso por VPN.

Tabla 5.1. IP address available on source code with RDP exposed

4.1.3 POSSIBLE OAS DEFAULT SOAP CONFIGURATION

Nivel	MEDIO
Descripción	El servicio SOAP en OAS (Oracle Application Server) parece tener una configuración por defecto.
Impacto	Un atacante podría utilizar esta configuración por defecto y subir código o leer información sensible del servidor.
Remediación	NO utilizar la configuración por defecto de SOAP.

Tabla 5.1. IP address available on source code with RDP exposed

5. RECOMENDACIONES FINALES

Luego de los resultados obtenidos en la auditoría se recomienda tomar las siguientes acciones:

1. Habilitar autenticación en los endpoints restantes (considerar activar el servicio MSS-TAS para estos sistemas)
2. Deshabilitar el servicio RDP.
3. Deshabilitar la configuración por defecto en el backend SOAP en OAS (Validar con white-box si existe este servicio).
4. Incluir a estos sistemas en el testeo de vulnerabilidades que se realiza semanalmente (servicio MSS-VME).
5. Considerar un servicio para detección de actividades sospechosas y contención de malware (en el ambiente de GLESEC es el MSS-EDR).