Kioptrix\_1

Análisis de Vulnerabilidades

Informe Técnico

Resumen

El día 04/03/2021 se entregó el informe de Análisis de Vulnerabilidades del proyecto llamado Kioptrix\_1 en la cual se han identificado cuatro vulnerabilidades de las cuales una es de riesgo crítico, tres son de riesgo alto, 0 son de riesgo medio y 0 son de riesgo bajo.

En el detalle de cada una de las vulnerabilidades se brinda una posible solución de las mismas y en el anexo encontrará a quien afecta.

El procedimiento utilizado es el de caja negra al no conocer nada más que la dirección ip y url proporcionadas por el cliente, siguiendo la metodología OSSTMM v 2.1, Manual de la Metodología Abierta de Testeo de Seguridad.

Introducción

En el presente documento se exponen los resultados de las vulnerabilidades detectadas en la aplicación después de realizar un Análisis de Vulnerabilidades.

Objetivo

Detectar las potenciales vulnerabilidades web en la aplicación Kioptrix\_1 y sugerir las posibles soluciones para los problemas detectados.

Vulnerabilidades detectadas

Se han detectado las siguientes vulnerabilidades:

* **SMBv2 exploit (CVE-2009-3103, Microsoft Security Advisory 975497) - Riesgo: Crítico**
* **Transport Layer Security (TLS) Protocol DHE\_EXPORT Ciphers Downgrade MitM (Logjam) - Riesgo: Alto**
* **SSL POODLE information leak - Riesgo: Alto**
* **SSL/TLS MITM vulnerability (CCS Injection) - Riesgo: Alto**

Se han encontrado cuatro vulnerabilidades de las cuales una es crítica y 3 vulnerabilidades son de riesgo alto.

**Web Analizadas:**

* <http://kioptrix1.casa.local>
* <https://kioptrix1.casa.local>

**IP Analizadas:**

* 192.168.240.87

Detalle de las vulnerabilidades

A continuación, se detallan las vulnerabilidades detectadas:

|  |
| --- |
| SMBv2 exploit (CVE-2009-3103, Microsoft Security Advisory 975497) - Riesgo: Crítico |
| **Descripción**  CVE:CVE-2009-3103  | Array index error in the SMBv2 protocol implementation in srv2.sys in Microsoft Windows Vista Gold, SP1, and SP2,  | Windows Server 2008 Gold and SP2, and Windows 7 RC allows remote attackers to execute arbitrary code or cause a  | denial of service (system crash) via an & (ampersand) character in a Process ID High header field in a NEGOTIATE  | PROTOCOL REQUEST packet, which triggers an attempted dereference of an out-of-bounds memory location,  | aka "SMBv2 Negotiation Vulnerability."  **Referencia**  https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3103  |\_ http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3103 |

|  |
| --- |
| Transport Layer Security (TLS) Protocol DHE\_EXPORT Ciphers Downgrade MitM (Logjam) - Riesgo: Alto |
| **Descripción**  CVE:CVE-2015-4000 BID:74733  | The Transport Layer Security (TLS) protocol contains a flaw that is  | triggered when handling Diffie-Hellman key exchanges defined with  | the DHE\_EXPORT cipher. This may allow a man-in-the-middle attacker  | to downgrade the security of a TLS session to 512-bit export-grade  | cryptography, which is significantly weaker, allowing the attacker  | to more easily break the encryption and monitor or tamper with  | the encrypted stream.  **Referencia**  https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2015-4000  | https://weakdh.org  | https://www.securityfocus.com/bid/74733 |

|  |
| --- |
| SSL POODLE information leak - Riesgo: Alto |
| **Descripción**  CVE:CVE-2014-3566 BID:70574  | The SSL protocol 3.0, as used in OpenSSL through 1.0.1i and other  | products, uses nondeterministic CBC padding, which makes it easier  | for man-in-the-middle attackers to obtain cleartext data via a  | padding-oracle attack, aka the "POODLE" issue.  **Referencia**  https://www.imperialviolet.org/2014/10/14/poodle.html  | https://www.securityfocus.com/bid/70574  | https://www.openssl.org/~bodo/ssl-poodle.pdf  |\_ https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566 |

|  |
| --- |
| SSL/TLS MITM vulnerability (CCS Injection) - Riesgo: Alto |
| **Descripción**  OpenSSL before 0.9.8za, 1.0.0 before 1.0.0m, and 1.0.1 before 1.0.1h  | does not properly restrict processing of ChangeCipherSpec messages,  | which allows man-in-the-middle attackers to trigger use of a zero  | length master key in certain OpenSSL-to-OpenSSL communications, and  | consequently hijack sessions or obtain sensitive information, via  | a crafted TLS handshake, aka the "CCS Injection" vulnerability.  **Impacto**  ssl-ccs-injection  **Referencia**  http://www.openssl.org/news/secadv\_20140605.txt  | http://www.cvedetails.com/cve/2014-0224  |\_ https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0224 |

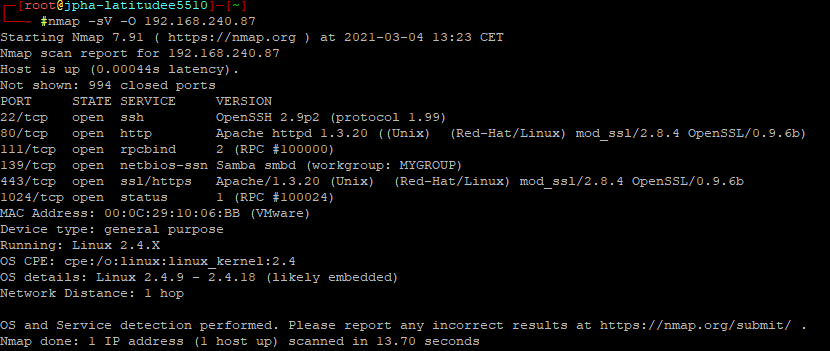
Conclusión

Se recomienda realizar un plan de acción para resolver los puntos expuestos en el presente informe.



Anexo 1. Técnicas empleadas.

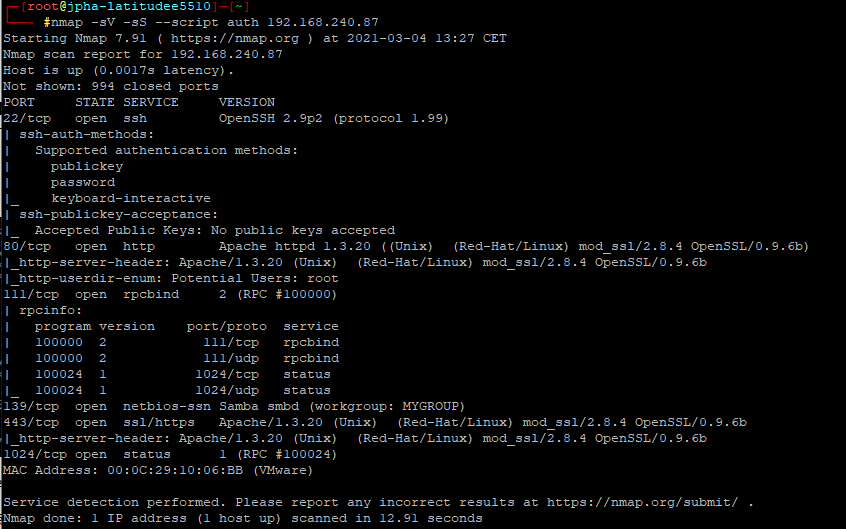
En primer lugar, se pasó la herramienta nmap para intentar descubrir sistema operativo, servicios y versiones:



Esto nos indica que tenemos un servidor Linux 2.4.x con los servicios:

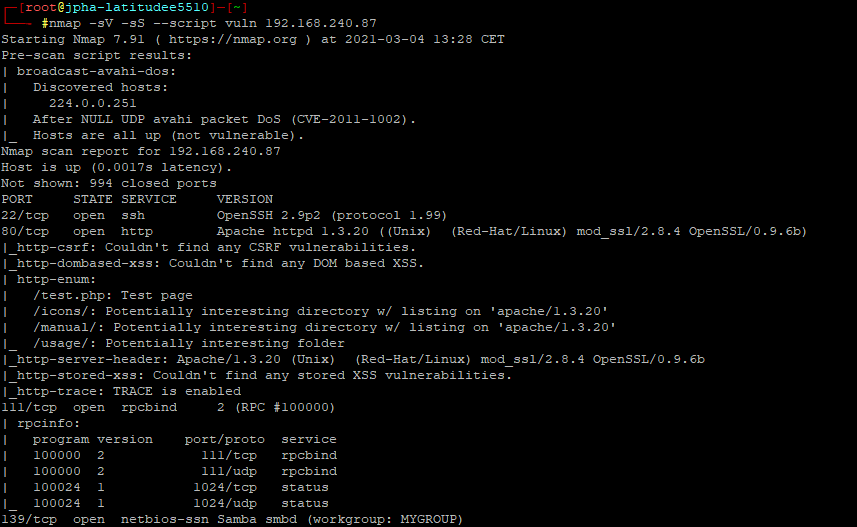
* Puerto 22 openssh 2.9p2
* Puerto 80 apache 1.3.20
* Puerto 111 rpc
* Puerto 139 smb
* Puerto 443 apache 1.3.20
* Puerto 1024

Ahora buscamos alguna brecha evidente, como usuarios y contraseñas por defecto:

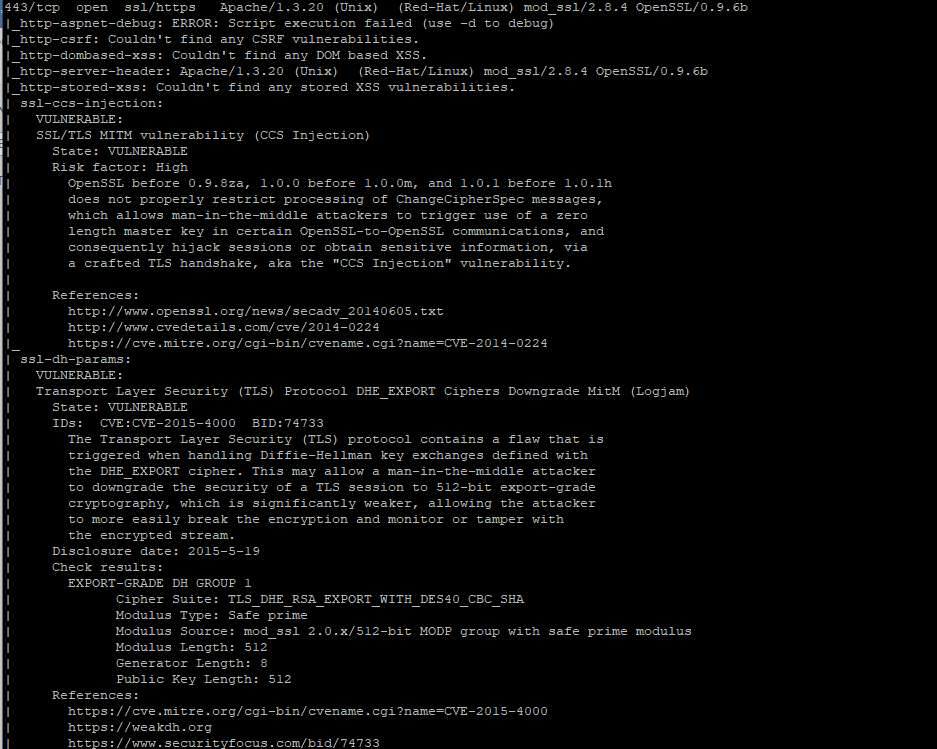


No parece tener nada importante.

Busquemos ahora vulnerabilidades con nmap:



Sigue:



Ahora sí aparecen posibles vulnerabilidades.

Como tenemos un servidor web, le pasamos la herramienta wfuzz:

Anexo 2. Informe Nessus.