Dom. Cien., ISSN: 2477-8818
Vol. 3, núm. mon., agos., 2017, pp. 505-516

Número Publicado el 22 de agosto de 2017

http://dx.doi.org/10.23857/dom.cien.pocaip.2017.3.mono1.ago.505-516
URL:http://dominiodelasciencias.com/ojs/index.php/es/index

Metodología OSSTMM para la detección de errores de seguridad y vulnerabilidad en sistemas operativos de 64 bits a nivel de usuario final

OSSTMM methodology for detecting security and vulnerability errors in 64-bit operating systems at the end user level

Metodologia OSSTMM para detectar erros de segurança e vulnerabilidade em sistemas operacionais de 64 bits no nível do usuário final

[I] Yolanda de la N. Cruz-Gavilanes
yolanda.cruz@cnt.gob.ec

[II] Carlos J. Martínez-Santander
carlos4553@hotmail.com

Recibido: 26 de enero de 2017 ***** **Corregido:** 13 de marzo de 2017 ***** **Aceptado:** 18 de julio de 2017

[I] Magíster en Seguridad Telemática, Ingeniera en Electrónica y Telecomunicaciones, Dibujante Técnica, Corporación Nacional de Telecomunicaciones.

[II] Magíster en Seguridad Telemática, Ingeniero de Sistemas, Catedrático de la Universidad Católica de Cuenca

Resumen

El objetivo de la investigación fue aplicar una metodología abierta de testeo de seguridad (OSSTMM) para la detección de errores de seguridad y vulnerabilidad en sistemas operativos de 64 Bits a nivel de usuario final, se recopiló información sobre vulnerabilidades más frecuentes que se presentan en los sistemas operativos Windows de 64 bits, además se experimentó, analizó y se aplicó el Manual de la metodología OSSTMM para solucionar estas vulnerabilidades en los equipos de cómputo, la aplicación de esta metodología se sustenta en cuatro fases de acuerdo a los requerimientos de la investigación como: 1) Levantamiento de la Información, 2) Análisis de vulnerabilidades en sistemas operativos, 3) Evaluación de riesgos, y 4) Capacitación al usuario. En los resultados se detectó un 95% de error de seguridad y de vulnerabilidades en los sistemas Windows de 64 bits que son cometidos por los usuarios finales por su desconocimiento en la configuración y actualizaciones de seguridad que se deben brindar a los sistemas informáticos de Windows de 64 bits, para evitar ser víctimas fáciles de los hackers para el robo y manipulación de equipos y datos. Por lo que se concluyó que los usuarios finales son los causantes de exponer a los equipos de cómputo a errores de seguridad por su ambiguo conocimiento en seguridad informática. Se recomienda para futuros trabajos continuar con la aplicación de la metodología OSSTMM para detectar los errores de seguridad en los sistemas informáticos y capacitar a los usuarios finales que utilizan equipos de cómputo.

Palabras clave: vulnerabilidad; Windows 64 bits; sistemas operativos; usuario final; metodología abierta de testeo de seguridad.

Abstract

The objective of this research work is to apply an Open Source Security Testing Methodology Manual (OSSTMM) for the detection of security and vulnerability errors of operative systems of 64 Bits at a final user level. The information about the most frequent vulnerabilities found in the operative system 64 Bits Windows was collected. Also, the OSSTMM was used, analyzed and applied in order to solve these vulnerabilities on computers. The application of this methodology is sustained on four phases according to the requirements of this work: 1) gathering information, 2) analysis of the vulnerabilities of the system, 3) evaluation of risks, and 4) user training. The results detected 95% of safety error and vulnerability of 64 Bit Windows. The final users, because of ignorance about configuration and safety actualizations, complete these errors. The computer 64 bit windows systems

should have update protection in order to avoid being hacked to steal or manipulate equipment and data. As a conclusion, it could be said that the final users are the ones who cause exposure of the equipment to safety errors and ambiguous knowledge on computer information safety. It is recommended to start applying OSSTMM to detect safety errors on computers and to train the final users to use the computer equipment correctly.

Keywords: vulnerability; Windows 64-bit; operating systems; end user; open security testing methodology.

Resumo

O objetivo da pesquisa foi aplicar uma metodologia de teste de segurança aberta (OSSTMM) para a detecção de erros de segurança e vulnerabilidade em sistemas operacionais de 64 bits no nível do usuário final, informações foram coletadas sobre as vulnerabilidades mais freqüentes que estão presentes em os sistemas operacionais Windows de 64 bits, o Manual de Metodologia OSSTMM também foi testado, analisado e aplicado para resolver essas vulnerabilidades em equipamentos informáticos, a aplicação desta metodologia baseia-se em quatro fases de acordo com os requisitos de pesquisa como 1) Pesquisa de informações, 2) Análise de vulnerabilidades em sistemas operacionais, 3) Avaliação de risco, y 4) Treinamento para o usuário. Os resultados detectaram um erro de 95% de segurança e vulnerabilidade nos sistemas Windows de 64 bits que são cometidos pelos usuários finais devido à falta de conhecimento nas atualizações de configuração e segurança que devem ser fornecidas aos sistemas informáticos do Windows de De 64 bits, para evitar ser fácil vítimas de hackers por roubo e manipulação de equipamentos e dados. Por conseguinte, concluiu-se que os utilizadores finais são responsáveis por expor o equipamento informático a erros de segurança devido ao seu conhecimento ambíguo na segurança informática. Recomenda-se que o trabalho futuro continue com a aplicação da metodologia OSSTMM para detectar erros de segurança em sistemas informáticos e para treinar usuários finais que usam equipamentos de informática.

Palavras chave: vulnerabilidade; Windows 64-bit; sistemas operacionais; usuário final; abra a metodologia de teste de segurança.

Introducción

En la época actual, el desarrollo de las tecnologías de la información ha tenido un salto colosal, la innovación y el crecimiento de las TIC en las organizaciones han repercutido en la mejora de los beneficios, tanto a nivel competitivo como eficiencia, por este mismo hecho se encuentran vulnerables, por cuanto se detectan situaciones de acceso no autorizado a los equipos de cómputo a través de la red, provocando la caída del sistema de forma esotérico, atacando en la confiabilidad, confidencialidad, y autenticad de los archivos que se encuentran en el equipo informático (Song, Hu, & Xu, 2009).

Por otra parte los hackers han aprovechado las configuraciones complejas que hay que realizar a un equipo informático para garantizar su seguridad, además se suma la rapidez con la que se actualiza la tecnología (Mora, 2005).

Los sistemas operativos deben ser valorados por expertos informáticos ya que representan una gran importancia para la información que va a ser almacenada; el sistema operativo es un software que ayuda en la interfaz entre usuario y ordenador (Salah, Calero, Bernabé, Perez, & Zeadally, 2013).

El sistema operativo es el elemento esencial de software de aplicaciones, sin este fundamento seguro las aplicaciones y los sistemas de seguridad no garantizarían la información almacenada (Yile, 2016).

Cuando se tiene entornos de red, la seguridad depende del sistema y de la configuración por parte del usuario que le dé a su computador, sin seguridad en los sistemas operativos no existirán valores confiables y afectará significativamente al sistema (Yile, 2016).

El propósito inicial de esta investigación es abordar las vulnerabilidades que tienen los sistemas operativos Windows de 64 bits en las versiones (XP, Vista, Seven, Eight, Server 2008) y concienciar a los usuarios finales que tienen que asegurar a su sistema informático para no ser víctima de hackeos o robos de información (Aziz & Sporea, 2014) (Liu et al., 2015).

El planteamiento surgió de la aparición de vulnerabilidades dentro de sistemas operativos, esto se debe a la inseguridad que brindan los propios usuarios en sus sistemas al no tener una metodología para detectar errores y problemas.

La investigación se enfocó en cómo reducir los errores de seguridad y las vulnerabilidades de sistemas operativos Windows de 64 bits a nivel de usuario final, a través de la utilización de una metodología OSSTMM (Herzog, P 2010, p 23).

El creador de la metodología OSSTMM lo definió como una guía para mejorar la seguridad en los equipos informáticos, es así que esta metodología se divide en canales, módulos, ambientes fases según sea la prueba de seguridad que se desea realizar (Herzog, P 2010, p 23). Internacionalmente, la metodología OSSTMM es estandarizada para las buenas prácticas de seguridad para implantación de un sistema de seguridad de información, (Franco, D & Guerrero, C 2013) todos estos canales.

Materiales y métodos

Esta investigación se basa en la utilización de un estudio descriptiva aplicativa fundamentada en la metodología OSSTMM.

La investigación descriptiva sirve para la recopilación de las diversas tendencias y los fundamentos del estudio de la reducción de la vulnerabilidad de seguridad en los sistemas operativos, y la investigación aplicativa permitirá generar criterios sobre la implementación de los diversos procedimientos de detección de errores, para reducir la vulnerabilidad de seguridad en los Sistemas operativos, tomando como fundamento la metodología OSSTMM.

Abarca dos de las áreas o canales que describe la tabla 1 que muestra:

Tabla 1. Canales y secciones de OSSTMM

CANAL	SECCION	DESCRIPCION
Seguridad Física	Humano	Elemento Humano
	Físico	Todo Objeto Tangible
Seguridad de las comunicaciones	Redes de Datos	Sistemas electrónicos y redes de datos
	Telecomunicaciones	Comunicaciones digitales o analógicas
Seguridad del espectro electromagnético	Comunicaciones inalámbricas	Incluyen las señales electromagnéticas

Fuente: (Valdez Alvarado, /)

La investigación se centra en dos canales de la metodología OSSTMM que ayudarán a una obtención eficaz de datos, el canal de seguridad físico con la sección humano y el canal seguridad de las telecomunicaciones con la sección redes de datos que tienen sus correspondientes tareas y procedimientos, de acuerdo al canal que está siendo evaluado (López, A., 2011). Además, hace referencia al canal humano debido a que se está trabajando directamente con usuarios finales, que están en directo contacto con un equipo de cómputo, ya que son el eslabón más débil de la seguridad.

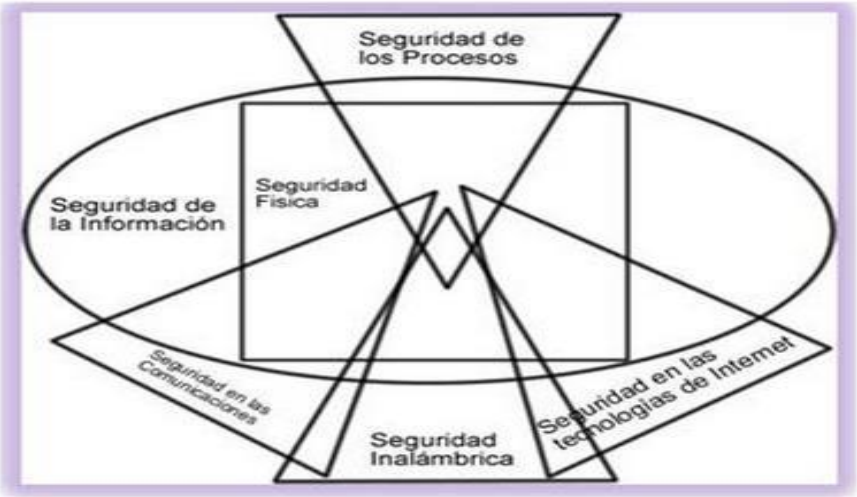

Figura 1 Esquema de la metodología OSSTMM
Fuente: (Prandini & Ramilli, 2010)

Con el fin de evaluar la seguridad de los sistemas operativos, se utiliza la metodología OSSTMM para describir las vulnerabilidades y referirse a sus precondiciones y estimaciones de ataques que puedan afectar el equipo de cómputo, ya que los hacker pueden valorar esa vulnerabilidad y realizar ataques a la información, y continuar con la siguiente vulnerabilidad que encuentre y sabotear todo el sistema.

Resultados

Para obtener los resultados de la investigación se ha evaluado cada uno de los sistemas operativos en estudio y así verificar a eficiencia y seguridad de cada uno de ellos, a través de la utilización de la metodología que se describe a continuación:

1) Levantamiento de información de los Sistemas Operativos de 64 bits:

Datos de seguridad que los usuarios finales han dado al equipo como la configuración de firewalls, contraseñas seguras, antivirus e información sobre el tipo de usuario que está a cargo del equipo informático como se observa en la figura 2.

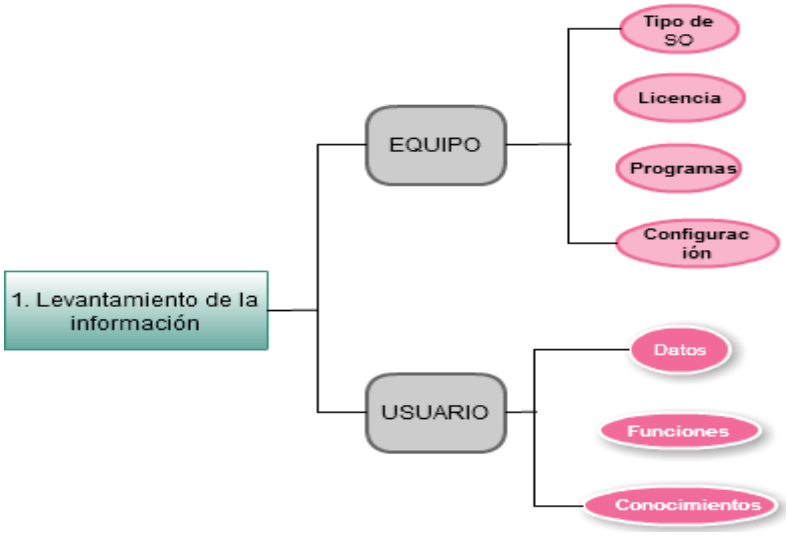

Figura 2. Levantamiento de la información

Fuente: (Cruz Yolanda, 2016)

2) Análisis de vulnerabilidades de Sistemas Operativos

Es donde se evalúan la seguridad del equipo y se recoge las vulnerabilidades que se obtienen al realizar un escaneo y realizar ataques intencionados.

Para observar el rendimiento y seguridad de cada uno de los sistemas operativos en estudio, se construyó tablas para cada versión para mostrar los problemas y soluciones de estimación de seguridad en las diferentes versiones de Windows (Xp, vista, Seven, Eight, y sever 2008).

Además, se aplicó a un cierto número de computadoras en una universidad del Austro, que por motivos de seguridad no es conveniente exponer los datos.

Tabla 2. Informe de vulnerabilidades de Windows Xp de 64 bits

SOLUCIÓN A LAS VULNERABILIDADES DEL SISTEMA OPERATIVO XP DE 64 BITS	
VULNERABILIDAD	**SOLUCIÓN**
Microsoft Server Service / CanonicalizePathName() Remote Code Execution Vulnerability (dcerpc-ms-netapinetpathcanonicalize-dos)	Instalar el parche desde http://download.microsoft.com/download/9/0/b/90b8dbba-09c1-4b27-b0c4-0cc13706823a/Windows2000-KB921883-x86-ENU.EXE
MS09-001: Vulnerabilities in SMB Could Allow Remote Code Execution (958687) (windows-hotfix-ms09-001)	Se instala el parche desde http://go.microsoft.com/fwlink/?LinkId=132991
MS10-012: Vulnerabilities in SMB Server Could Allow Remote Code Execution (971468) (windows-hotfix-ms10-012)	Instalar el parche desde http://go.microsoft.com/fwlink/?LinkId=155976
MS10-054: Vulnerabilities in SMB Server Could Allow Remote Code Execution (982214) (windows-hotfix-ms10-054	Instalar el Parche desde http://go.microsoft.com/fwlink/?LinkId=190318
MS11-020: Vulnerability in SMB Server Could Allow Remote Code Execution (2508429) (windows-hotfix-ms11-020)	Instalar el parche desde http://go.microsoft.com/fwlink/?LinkId=212236
CIFS NULL Session Permitted (cifs-nt-0001)	Configurar Microsoft Knowledge Base Article Q246261
MS06-035: Vulnerability in Server Service Could Allow Remote Code Execution (917159) (windows-hotfix-ms06-035)	Instalar parche desde http://go.microsoft.com/fwlink/?LinkId=64331
SMB signing disabled (cifs-smb-signing-disabled)	Configurar this TechNet article
SMB signing not required (cifs-smb-signing-not-required	Configurar this TechNet article
ICMP timestamp response (generic-icmp-timestamp	Deshabilitar timestamp ICMP a partir del comando de control firewalls de Windows
NetBIOS NBSTAT Traffic Amplification (netbios-nbstat-amplification)	Restringiendo el acceso a los netBios solo a servicios activos de confianza

Fuente:(Cruz Yolanda, 2016)

La tabla 2 muestra las vulnerabilidades que presenta esta versión de Windows Xp, ratificando que existe una debilidad en la seguridad del sistema operativo y en la configuración por parte del usuario final, pero a la vez presenta las soluciones que el usuario puede utilizar o configurar su equipo.

Tabla 3. Informe de vulnerabilidades de windows vista de 64 bits

SOLUCIÓN DE VULNERABILIDADES DEL SISTEMA OPERATIVO VISTA DE 64 BITS	
VULNERABILIDAD	**SOLUCIÓN**
MS09-050: Vulnerabilities in SMBv2 Could Allow Remote Code Execution (975517) (windows-hotfix-ms09-050)	Instalar parche desde http://go.microsoft.com/fwlink/?LinkId=163970
MS10-054: Vulnerabilities in SMB Server Could Allow Remote Code Execution (982214) (windows-hotfix-ms10-054)	Instalar parche desde http://go.microsoft.com/fwlink/?LinkId=190318
MS11-020: Vulnerability in SMB Server Could Allow Remote Code Execution (2508429) (windows-hotfix-ms11-020)	Aplicar parche desde http://go.microsoft.com/fwlink/?LinkId=212236
SMB signing disabled (cifs-smb-signing-disabled)	Configurar a partir this TechNet article
SMB signing not required (cifs-smb-signing-not-required)	Restringiendo el acceso a los netBios solo a servicios activos de confianza
ICMP timestamp response (generic-icmp-timestamp)	Deshabilitar ICMP timestamp en el panel de control
TCP timestamp response (generic-tcp-timestamp)	Deshabilitar TCP
NetBIOS NBSTAT Traffic Amplification (netbios-nbstat-amplification)	Restringiendo el acceso a los netBios solo a servicios activos de confianza

Fuente: (Cruz Yolanda, 2016)

La tabla 3 presenta las vulnerabilidades y soluciones de la versión de Windows vista, y se observa que se tomó más en cuenta la seguridad; los usuarios deberían tomar en cuenta la configuración de algunos parámetros.

Tabla 4. Informe de vulnerabilidades de Windows server 2008 de 64 bits

SOLUCIÓN DE VULNERABILIDADES DEL SISTEMA OPERATIVO WINDOWS SERVER 2008 DE 64 BITS	
VULNERABILIDAD	**SOLUCIÓN**
SMB signing disabled (cifs-smb-signing-disabled)	Configurar this TechNet article
SMB signing not required (cifs-smb-signing-not-required)	Configurar this TechNet article
ICMP timestamp response (generic-icmp-timestamp)	Deshabilitar ICMP timestamp en el panel de control
TCP timestamp response (generic-tcp-timestamp)	Deshabilitar TCP
NetBIOS NBSTAT Traffic Amplification (netbios-nbstat-amplification)	Restringiendo el acceso a los netBios solo a servicios activos de confianza

Fuente:(Cruz Yolanda, 2016)

Tabla 5. Informe de vulnerabilidades de Windows seven de 64 bits

VULNERABILIDADES DEL SISTEMA OPERATIVO WINDOWS SEVEN DE 64 BITS	
VULNERABILIDAD	**DESCRIPCIÓN**
SMB signing disabled (cifs-smb-signing-disabled)	Configurar this TechNet article
SMB signing not required (cifs-smb-signing-not-required)	Configurar this TechNet article
TCP timestamp response (generic-tcp-timestamp)	deshabilitar TCP desde panel de control
UPnP SSDP Traffic Amplification (upnp-ssdp-amplification)	Restringir el acceso a la función UPnP para activos solamente de confianza
NetBIOS NBSTAT Traffic Amplification (netbios-nbstat-amplification)	Restringiendo el acceso a los netBios solo a servicios activos de confianza

Fuente:(Cruz Yolanda, 2016)

Tabla 4 y tabla 5, muestran que suma menos vulnerabilidades que la versión anterior, pero aun así continúan la inseguridad en los sistemas operativos por el desconocimiento o falta de precaución por parte del usuario, sus propias tablas dan las posibles soluciones.

Tabla 6. Informe de vulnerabilidades de Windows Eight de 64 bits

VULNERABILIDADES DEL SISTEMA OPERATIVO WINDOWS EIGHT DE 64 BITS	
VULNERABILIDAD	**DESCRIPCIÓN**
NetBIOS NBSTAT Traffic Amplification (netbios-nbstat-amplification)	Restringiendo el acceso a los netBios solo a servicios activos de confianza

Fuente: (Cruz Yolanda, 2016)

La tabla 6, demuestra que Windows 8 funciona mejor que las versiones anteriores en lo que a seguridad se refiere, entonces la suma de riesgos de vulnerabilidades es menor, pero a pesar de existir una mínima cantidad, la no configuración por parte del usuario final sería la puerta segura para el ingreso de las atacantes, y robar o manipular la información.

Para comprobar el desconocimiento en seguridad de sistemas operativos, fue necesario la aplicación de la metodología en una importante universidad del Austro que por motivos de seguridad no es posible poner datos de la misma, además se usó un programa estadístico como es SPSS versión 22 con licencia; se aplicó a 40 sistemas operativos en donde independientemente del sistema operativo se anotó el número de vulnerabilidades, y al final se aplicó una prueba z que dio como resultado con un índice de confianza del 95% por tanto un margen de error del 5%, se presenta la comprobación de la hipótesis en el siguiente cuadro.

Tabla 7. Estadística de muestra única

Estadísticas de muestra única

	N	Media	Desviación estándar	Media de error estándar
Nu_Vulnerabilidad	40	4,13	2,323	,367

Fuente: (Cruz Yolanda, 2016)

Tabla 8. Prueba de muestra única

Prueba para una muestra

	Valor de prueba = 4					
	t	gl	Sig. (bilateral)	Diferencia de medias	95% Intervalo de confianza para la diferencia	
					Inferior	Superior
Nu_Vulnerabilidad	,340	39	,735	,125	-,62	,87

Fuente: (Cruz Yolanda, 2016)

Conclusiones

-En este trabajo se analizó las vulnerabilidades de las distintas versiones de Windows que son causadas por los usuarios finales. En particular, para su evaluación de seguridad se utilizó la metodología OSSTM y se realizó unas tablas para cada una de las versiones y se extrajeron las vulnerabilidades y las posibles soluciones que pueden utilizar los usuarios finales, ya que el principal problema de seguridad seguirá siendo el usuario.

-La comparación de las versiones de Windows dio como resultado que la seguridad está más implementada en Windows eight con menos vulnerabilidades que las anteriores.

-Los usuarios finales son el eslabón más débil de la seguridad informática.

- Lo que se detectó en el análisis, es que la institución no le da importancia a la seguridad de los sistemas operativos ya que piensan que es una pérdida de recursos.

Referencias bibliográficas

Aziz, B., & Sporea, I. (2014). Security and VO management capabilities in a large-scale Grid operating system. Computing and Informatics, 33(2), 303-326.

Cruz Yolanda. (2016, marzo). Metodología OSSTMM para la detección de errores de seguridad y vulnerabilidad e n sistemas operativos de 64 bits a nivel de usuario final. Escuela Superior Politecnica de Chimorazo, Riobamba.

Liu, K., Tian, M., Liu, T., Jiang, J., Ding, Z., Chen, Q., Zhang, X. (2015). A high-efficiency multiple events discrimination method in optical fiber perimeter security system. Journal of Lightwave Technology, 33(23), 4885-4890.

Mora, L. (2005). Niveles de seguridad lógica contra ataques externos a través de internet en una plataforma Windows 2000 server en empresas de tecnología. Télématique: Revista Electrónica de Estudios Telemáticos, 4(2), 44-59.

Prandini, M., & Ramilli, M. (2010). Towards a practical and effective security testing methodology. En Computers and Communications (ISCC), 2010 IEEE Symposium on (pp. 320-325). IEEE.

Salah, K., Calero, J. M. A., Bernabé, J. B., Pérez, J. M. M., & Zeadally, S. (2013). Analyzing the security of Windows 7 and Linux for cloud computing. Computers & security, 34, 113-122.

Song, J., Hu, G., & Xu, Q. (2009). Operating system security and host vulnerability evaluation. En Management and Service Science, 2009. MASS'09. International Conference on (pp. 1-4). IEEE.

Valdez Alvarado, A. (/). OSSTMM 3. Revista de Información, Tecnología y Sociedad, 29.

Yile, F. (2016). Research on the Security Problem in Windows 7 Operating System. En Measuring Technology and Mechatronics Automation (ICMTMA), 2016 Eighth International Conference on (pp. 568-571). IEEE.