

Tarea 1 - Reporte de L^AT_EX

Kathy Brenes Guerrero, Barnum Castillo Barquero

Maestría en Ciencias de la Computación, Introducción a la Investigación, ITCR

Abstract—One of the biggest issues that an operating system can experience is privilege escalation. Privilege escalation is the act of exploiting a bug, design flaw, or configuration oversight in an operating system or software application to gain elevated access to resources that are normally protected from an application or user. Understanding the weaknesses and flaws of a security level issue for the operating system can help implement better approaches and techniques to improve the software itself. Just because you have updated your computer to the latest update or patch, doesn't mean that it has been secured. Windows, for example, has a series of vulnerabilities that can affect the operating system and can't be solved by Microsoft because the updates can create incompatibilities with an older system or with some security protocols. The Privilege Escalation technique takes advantage of these vulnerabilities to gain privileges (access) within a remote system, in order to run applications and make commands on it. The focus of this paper is to list the vulnerabilities that have been demonstrated by third party systems in different operating system, and provide a technical point of view on what can be done to avoid these breaches (vulnerabilities or impacts). An Operation System breach can enable attackers to increase their level of control over target systems, such that they are free to access any data or make any configuration changes. This study reveals the importance of the way in which current systems should be defended from this mechanism.

Index Terms—Operating System, Penetration Testing, Cyber-security, Internet of Things.

I. INTRODUCCIÓN

El siguiente reporte detalla los conceptos básicos y principales del lenguaje de programación L^AT_EX, además de su historia. Cada una de las siguientes secciones incorpora un ejemplo aplicado de los términos comentados.

II. HISTORIA DE L^AT_EX

TeX es el programa original de composición matemática desarrollado alrededor de 1980 por Donald Knuth para la composición tipográfica digital de alta calidad.[?] TEX es un lenguaje de bajo nivel con el que las computadoras pueden trabajar, pero a la mayoría de las personas les resultaría difícil usarlo; entonces L^AT_EX ha sido desarrollado para hacerlo más fácil.[?]. Se pronuncia como "tecnología" como en alta tecnología. La X es la letra griega chi, que hace que el sonido "ch" aparezca al final de "tech". En la documentación original para TeX, hay mucha discusión acerca de "pegar" objetos juntos y de "estirar" el espacio entre los objetos que componen una página. [?]

El látex es un producto natural pegajoso que forma la base del caucho, y creo que ese es el motivo de la palabra L^AT_EX. L^AT_EX está construido sobre TeX. Fue escrito a principios de

la década de 1980 por Leslie Lamport. Tiene más funciones de alto nivel integradas que TeX, por lo que tiende a ser más fácil de usar. L^AT_EX no es un intento de sonar en francés; probablemente no sea La TeX, o "The TeX". La razón de la divertida alternancia de letras mayúsculas y minúsculas es que TeX y L^AT_EX, cuando se escriben correctamente de esta manera:, son todas letras mayúsculas pero en diferentes tamaños de puntos y diferentes alturas por encima y por debajo de la línea de base. Alternando mayúsculas y minúsculas imita esto. [?]

L^AT_EX se creó para facilitar la producción de libros y artículos de uso general dentro de TeX. Debido a que L^AT_EX es una extensión del sistema de composición tipográfica TeX, tiene la capacidad de TeX para compilar documentos técnicos que contienen ecuaciones matemáticas complejas. Esta característica hizo que L^AT_EX fuera popular entre científicos e ingenieros. [?]

La producción de un documento L^AT_EX comienza con un archivo de texto que contiene contenido etiquetado con códigos especiales L^AT_EX utilizados para indicar cómo se diseñará el texto. Cuando el archivo se ejecuta a través de un procesador L^AT_EX, se producen páginas de composición tipográfica. Debido a que la composición tipográfica L^AT_EX requiere envolver el texto en códigos informáticos complicados, tiene una curva de aprendizaje bastante empinada. Aunque ahora hay programas de software que ayudan a automatizar la creación de documentos L^AT_EX, un conocimiento práctico de L^AT_EX sigue siendo deseable para este tipo de composición tipográfica.

L^AT_EX fue uno de los primeros programas de composición tipográfica capaz de producir ecuaciones matemáticas complejas. Con los años se ha utilizado para componer muchas revistas científicas, matemáticas y de ingeniería. La American Mathematical Society (AMS) incluso tiene su propio conjunto de extensiones, llamado AMS-L^AT_EX, que sus contribuyentes usan para su revista. Pero los programas de autoedición como Quark Inc. de Quark Inc. y FrameMaker de Adobe Systems Incorporated se volvieron más capaces de producir expresiones matemáticas complejas, L^AT_EX se hizo menos popular.[?] La versión actual de L^AT_EX es L^AT_EX2e. [?]

III. USOS ACADÉMICOS, EXTENSIÓN, IMPORTANCIA

L^AT_EX es un sistema de preparación de documentos para producir documentos de aspecto profesional, no es un procesador de textos. Es particularmente adecuado para producir documentos largos y estructurados, y es muy bueno para escribir ecuaciones. Está disponible como software libre para

la mayoría de los sistemas operativos.

Si está acostumbrado a producir documentos con Microsoft Word, encontrará que \LaTeX es un estilo de trabajo muy diferente. Microsoft Word es 'Lo que ves es lo que obtienes' (WYSIWYG), esto significa que puedes ver cómo se verá el documento final mientras escribes. Cuando trabaje de esta manera, probablemente realice cambios en la apariencia del documento (como espacios entre líneas, encabezados, saltos de página) mientras escribe. Con \LaTeX no verá cómo se verá el documento final mientras lo está escribiendo; esto le permite concentrarse en el contenido más allá de la apariencia. Para producir esto en la mayoría de los sistemas de tipografía o procesamiento de textos, el autor debería decidir qué diseño usar, por lo que seleccionaría (digamos) 18pt Times Roman para el título, 12pt Times Italic para el nombre, y así sucesivamente. Esto tiene dos resultados: los autores pierden su tiempo con los diseños; y muchos documentos mal diseñados. [?] \LaTeX se basa en la idea de que es mejor dejar el diseño del documento a los diseñadores de documentos y permitir que los autores continúen con la escritura de documentos. [?]

Un documento \LaTeX es un archivo de texto sin formato con una extensión de archivo .tex. Se puede escribir en un editor de texto simple como el Bloc de notas, pero la mayoría de las personas encuentran que es más fácil usar un editor de \LaTeX dedicado. Mientras escribe, marque la estructura del documento (título, capítulos, subtítulos, listas, etc.) con etiquetas. Cuando finaliza el documento, compílelo; esto significa convertirlo a otro formato. [?]

Existen varios formatos de salida diferentes, pero probablemente el más útil sea Portable Document Format (PDF), que aparece tal como se imprimirá y se puede transferir fácilmente entre computadoras. [?]

Implementaciones de \LaTeX

- 1) Composición de artículos de revistas, informes técnicos, libros y presentaciones de diapositivas.
- 2) Control sobre documentos grandes que contienen secciones, referencias cruzadas, tablas y figuras.
- 3) Composición tipográfica de fórmulas matemáticas complejas.
- 4) Composición tipográfica avanzada de las matemáticas con AMS- \LaTeX .
- 5) Generación automática de bibliografías e índices.
- 6) Composición tipográfica multilingüe.
- 7) Inclusión de obras de arte, y color de proceso o mancha.
- 8) Utilizando fuentes PostScript o Metafont.

A. Tipo de archivo

Los archivos \LaTeX no son tan self-contained como, por ejemplo, los documentos de Word. Sin embargo, sus tamaños de archivo son generalmente más pequeños. Aquí hay una descripción muy básica.

Un archivo fuente \LaTeX debe recibir la extensión .tex. Es un archivo de texto, uno que puede editar con Notepad o SimpleText, a diferencia de los documentos de Word, que solo puede editar con Word. El archivo fuente \LaTeX contiene el texto de sus documentos más los comandos que indican cómo deben formatearse el texto y las ecuaciones. De esta manera,

es un poco como código de computadora o HTML sin formato. Los archivos fuente \LaTeX deben ser procesados por un programa que sepa cómo interpretar los diversos comandos integrados en ellos. En una PC, puede descargar varias implementaciones de \LaTeX , como MiKTeX. En Macintosh, puede usar OzTeX. En una máquina UNIX, debería ser capaz de instalar \LaTeX . Me referiré genéricamente a estos programas como \LaTeX . Tenga en cuenta que el programa texify puede ser el programa real que se utiliza para procesar un documento. Cuando latex procesa un archivo fuente \LaTeX llamado example.tex, crea un archivo llamado example.dvi. dvi significa Device Independent. Es un formato anterior a .pdf, y pretendía ser lo que dice, un formato que pueden manejar muchos dispositivos diferentes (plataformas de computadora). Cada plataforma de computadora tiene sus propios programas que pueden ver (algunos dicen vista previa) archivos .dvi. En una PC, MikTeX viene con YAP (Yet Another Previewer), en Macintosh, una vista previa dvi está integrada en OzTeX, y en una máquina UNIX, generalmente puede encontrar un programa llamado xdvi. latex también produce un archivo .aux y .log, del cual no tenemos que preocuparnos ahora.

La progresión habitual al componer un documento en \LaTeX es escribir el archivo .tex usando un editor de texto, luego procesarlo usando latex, luego ver el resultado de .dvi usando un reproductor de vista previa. Comparado con Word, esto parece bastante torpe, lo sé. Se debe principalmente al hecho de que \LaTeX se desarrolló en una plataforma UNIX, y así es como los programas UNIX tienden a estructurarse. Porque es más modular y más fácil tener a muchas personas involucradas en el desarrollo, en lugar de una sola empresa monolítica de desarrollo de software. En balance, eso es algo bueno.

No puede enviar un archivo .dvi a una impresora. La mayoría de las impresoras aceptan entradas PostScript (.ps). En una PC, puede abrir el archivo .dvi con YAP para imprimirlo. Con OzTeX, puede seleccionar Archivo, Imprimir. En UNIX, primero utiliza un programa como dvips para convertir el archivo .dvi a un archivo .ps, luego envía el archivo .ps a la impresora usando un comando como lp file.ps.

Finalmente, una vez que crea un documento \LaTeX , puede compartir el resultado electrónicamente con otros, ya sea por correo electrónico como un archivo adjunto o por publicarlo en una página web. Las personas que no usan TeX o \LaTeX generalmente no tienen instaladas las previsualizaciones .dvi y .ps, pero a menudo tienen Adobe Acrobat Reader, que puede leer y visualizar archivos .pdf (Portable Document Format). En una PC, probablemente use pdflatex para procesar el código fuente de \LaTeX y producir salida .pdf automáticamente. O puede usar un convertidor de DVI a PDF como dvipdfm, que viene con MiKTeX. En un Macintosh, puede usar Adobe Distiller (use Sherlock para buscarlo, luego ejecútelo) para convertir un archivo .ps en un archivo .pdf, luego use Acrobat Reader para verlo. En UNIX, puede usar dvipdf o ps2pdf. La principal desventaja de producir directamente salida .pdf de \LaTeX al editar su archivo fuente .tex es que una vez que Adobe Reader tiene un archivo abierto, latex no puede modificar el archivo en el disco. Las previsualizaciones dvi como YAP no tienen este problema.

IV. ESTILOS DE DOCUMENTO

```
\documentclass[options]{article}
Preamble (for LaTeX commands only)
\begin{document}
Document text (text with embedded LaTeX
commands)
\end{document}
```

Mediante el comando **document class** se determina el diseño y la estructura general del documento. Además la opción **article** permite establecer específicamente los usos y las propiedades que se van a desarrollar, otras clases comúnmente utilizadas son:

A. Article:

como su nombre indica, destinada a escribir artículos. Esto significa documentos relativamente cortos que no contienen capítulos o partes, solo secciones, subsecciones, etc. Como una de las clases base, el formato es bastante básico. Sin embargo, como la clase de artículo proporciona la función básica que la mayoría de la gente espera de LaTeX, a menudo se usa con modificaciones para documentos más largos.

```
\documentclass[opciones]{article}
Preamble (declaraciones : paquetes,
comandos; ttulo, autor, fecha)
\begin{document} Documento
\maketitle
\begin{abstract} ...
\end{abstract}
\section{... }
\subsection{... }
\subsubsection{...}
\end{document}
```

Comandos importantes

- 1) `\maketitle` Hace que se produzcan las líneas para el título, autor y fecha. Debe ubicarse después de `\begin{document}`, si se omite, no se generan dichos campos.
- 2) `\date` Se imprime la fecha vigente del computador, o el valor que se ingrese al campo obligatorio, si se desea que no aparezca de debe escribir `\date`.
- 3) `\thanks{...}` Se puede utilizar en `\title`, `\author`, `\date`, produce notas al pie de página con la información del autor.
- 4) `\begin{abstract}...\end{abstract}` En este entorno se coloca el resumen del artículo y debe ubicarse después de `\maketitle`.
- 5) `\section`, `\subsection` Son numeradas automáticamente.

B. Report:

La clase de informe está destinada a documentos más largos que tendrán capítulos, mientras que el libro está destinado a documentos muy grandes. La configuración estándar para informe y libro es ligeramente diferente de artículo. Por ejemplo, el valor predeterminado para el artículo es poner la información de `\maketitle` en la parte superior de la

primera página, mientras que el informe y el libro usan páginas de título separadas. libro incluye accesos directos predefinidos para `\frontmatter` (capítulos sin numerar con números de página romanos), `\mainmatter` (capítulos numerados y números de página arábica) y `\backmatter`. Tiene la misma estructura de book. Imprime por una sola cara: `\oneside`. El entorno `abstract` esta disponible, se crea en una página independiente.

C. Thesis:

Para escribir una tesis de RPI.

D. Book:

Para la estructura, formato y composición de libros.

```
\documentclass{book}
Preamble (declaraciones : paquetes, comandos;
título, autor, fecha)
\begin{document}
\maketitle
\frontmatter
\mainmatter
\chapter{...}
\section{...}
\subsection{...}
\appendix
\backmatter
\end{document}
```

Comandos importantes

- 1) `\frontmatter`: Apertura del libro, se presenta todo aquel contenido que no tenga que ver con el tema central tratado en el libro: prólogo, agradecimientos, tabla de contenido, derechos de autor, índice de figuras, índice de tablas etc. La numeración se realiza utilizando numeración romana.
- 2) `\mainmatter` Contiene la parte central del documento, se desarrolla el tema tratado en el libro, también se ubican los apéndices mediante el comando `\appendix` los cuales se numeran con las letras mayúsculas A, B, C,...
- 3) `\backmatter` Es el cierre del documento, contiene el índice alfabético, bibliografía, conclusiones, reconocimientos, información editorial, etc. Los capítulos no son numerados

E. Letter:

Para redactar el formato de una carta.

```
\documentclass{letter}
\usepackage[spanish]{babel}
\usepackage[latin1]{inputenc}
\begin{document}
\address{
Revista Colombiana de Estadística \\\
Universidad Nacional de Colombia}
\signature{
Pepito Prez \\\
Editor}
\date{30 septiembre de 2006}
\begin{letter}{
Dr. Donald Knuth \\\
```

```
Departamento de Ciencias de la Computación \\  
Universidad de Stanford \\  
EEUU  
}
```

F. Slides:

Para el diseño y confección de las diapositivas elaboradas con Beamer. El entorno slides da lugar a una transparencia, la numeración es consecutiva en la parte inferior derecha. Slides no se utilizan `\chapter`, `\section`, `\pagestyle`, `\thispagestyle`, ni los entornos `table` y `figure`. Los demás comandos se pueden utilizar libremente.

```
\documentclass{slides}  
Preamble (declaraciones : paquetes, comandos;  
    ttulo, autor, fecha)  
\begin{document}  
\begin{slides}  
Contenido de la transparencia  
\end{slides}  
\end{document}
```

Entre otras cosas, las clases proporcionan comandos de encabezado, como `\ part`, `\ chapter`, `\ section`

```
\documentclass[options]{article}  
Preamble (for LaTeX commands only)  
\begin{document}  
Document text (text with embedded LaTeX  
    commands)  
\end{document}
```

V. USO DE ELEMENTOS

A. Párrafos

Text here..

B. Efectos de letra

Text here..

C. Tildes

Text here..

D. Títulos

Text here..

E. Subtítulos

Text here..

F. Referencias

Text here..

G. Marcas de agua

Text here..

H. Headers y Footers

Text here..

I. Manejo de saltos de página

Text here..

J. Columnas de la página

Text here..

VI. MANEJO DE TABLAS

A. Subsection 1

Text here..

VII. MANEJO DE FIGURAS Y GRÁFICOS

A. Subsection 1

Text here.

VIII. MANEJO DE FIGURAS AL LADO DE TABLAS (MINIPAGE)

A. Subsection 1

Text here.

IX. ECUACIONES MATEMÁTICAS

A. Subsection 1

Text here.

X. MANEJO DE COLORES

A. Subsection 1

Text here.

REFERENCES

- [1] The Editors of Encyclopaedia Britannica (2013) *LaTeX COMPUTER PROGRAMMING LANGUAGE* [Blog post]. Consultado desde <https://www.britannica.com/technology/LaTeX-computer-programming-language>
- [2] *Introduction to LaTeX*. (2018). Consultado desde <https://www.latex-project.org/about/>
- [3] *Basic description of file types and how LaTeX works*. (2018). Consultado desde http://personal.bgsu.edu/~zirbel/5920/latex/latex_basics.htm