INSTITUTO POLITÉCNICO NACIONAL

“ESCOM” (ESCUELA SUPERIOR DE CÓMPUTO)

VÁZQUEZ BLANCAS CÉSAR SAID

U.A.: TEORIA DE LA COMPUTACION

GRUPO: 4CM4

Vulnerability Assessment in Heterogeneous Web Environment Using Probabilistic Arithmetic Automata

12/05/2024

INTRODUCCION

En la era digital actual, las empresas dependen en gran medida de las aplicaciones web heterogéneas para sus operaciones diarias. Estas aplicaciones, que combinan componentes como HTTP y servicios web SOAP o REST, ofrecen una amplia gama de funcionalidades pero también presentan desafíos significativos en términos de seguridad. A pesar de los esfuerzos por parte de organizaciones como OWASP para identificar y mitigar las vulnerabilidades más comunes en estas aplicaciones, la naturaleza dinámica y global de Internet las hace propensas a una variedad de ataques que pueden explotar estas debilidades de seguridad.

El presente trabajo se centra en abordar esta problemática mediante un enfoque proactivo y predictivo utilizando Automatas Aritméticas Probabilísticas (PAA). Si bien existen herramientas de detección de vulnerabilidades disponibles en el mercado, muchas de ellas carecen de la capacidad de adaptarse rápidamente a las actualizaciones y de identificar nuevas vulnerabilidades emergentes. Esta falta de adaptabilidad y proactividad deja a las empresas en riesgo de sufrir ataques que podrían tener consecuencias graves en términos de pérdida de datos, daño a la reputación y pérdidas financieras.

Problema y Contribuciones

El principal problema que aborda este trabajo es la necesidad de herramientas más proactivas y consistentes para detectar y prevenir vulnerabilidades en entornos web heterogéneos. Aunque existen herramientas disponibles para monitorear y detectar vulnerabilidades existentes, estas a menudo no pueden adaptarse fácilmente a las actualizaciones o identificar nuevas amenazas emergentes. Esto deja a las empresas en un estado de vulnerabilidad constante, sin la capacidad de prever y mitigar eficazmente los ataques.

Las contribuciones de este trabajo son múltiples y significativas:

1.-Propuesta de un modelo predictivo: Se presenta un modelo predictivo que utiliza Automatas Aritméticas Probabilísticas para evaluar y predecir la probabilidad de ocurrencia de ataques en aplicaciones web heterogéneas. Este enfoque permite una detección más temprana de posibles vulnerabilidades y una respuesta más proactiva ante amenazas emergentes.

2.-Incorporación de técnicas de Machine Learning: Se emplean técnicas de Machine Learning, específicamente Automatas Aritméticas Determinísticas y Probabilísticas, para mejorar la precisión y eficacia en la predicción de ataques y sus causas subyacentes. Esto permite una adaptación más rápida a los cambios en el panorama de amenazas y una mayor capacidad para identificar patrones de ataque complejos.

3.-Análisis de la penetración de ataques: Se analiza la capacidad de los ataques para penetrar desde aplicaciones web hasta servicios web, lo que proporciona información

valiosa sobre la extensión y el impacto potencial de los ataques. Esto permite a las empresas comprender mejor sus puntos vulnerables y tomar medidas proactivas para mitigar riesgos.

4.-Abogar por un modelo estándar: Se destaca la necesidad de un modelo estándar para detectar todas las posibles vulnerabilidades en aplicaciones web heterogéneas. Además, se aboga por la automatización completa del proceso de detección y la mejora en la presentación de resultados para facilitar la verificación y comprensión por parte de los desarrolladores y analistas de seguridad.

DESARROLLO

El sistema propuesto para la detección y prevención de vulnerabilidades en entornos web heterogéneos se fundamenta en una arquitectura robusta y proactiva. A continuación, se detalla cada componente de esta arquitectura, incluyendo el funcionamiento de los Automatas Aritméticas Probabilísticas (PAA) que son la columna vertebral de esta solución.

1. Recolección de Datos: La primera etapa del proceso implica la recolección de datos procedentes de diversas herramientas de análisis de seguridad web, tales como soap UI, rest UI, Burp suite, soap sonar. Estas herramientas proporcionan secuencias de datos que representan las interacciones con las aplicaciones web.

2. Análisis de Secuencias: Una vez recolectadas las secuencias de datos, estas son sometidas a un análisis exhaustivo. Durante esta etapa, se identifican patrones y se evalúan posibles vulnerabilidades en función de esos patrones. Se emplean patrones de ataques conocidos como referencia para este análisis.

3. Automatas Aritméticas Probabilísticas (PAA): Los PAA son fundamentales en esta arquitectura para modelar el comportamiento de las secuencias de datos y evaluar la probabilidad de ocurrencia de ataques. Estos automatas constan de estados, transiciones, valores y emisiones.

A. AUTOMATAS ARITMÉTICAS DETERMINÍSTICAS

Para modelar cálculos deterministas en secuencias, definimos una contraparte determinista para las PAAs. Considerando el escenario de evaluación de vulnerabilidades, se puede definir una tupla de autómata aritmético determinista como:

$$A=\{Q,\Sigma,\delta,E,V0,v0,q0,(\mu q)q\in Q,(\lambda q)q\in Q\}A=\{Q,\Sigma,\delta,E,V0\ ,v0\ ,q0\ ,(\mu q\ )q\in Q\ ,(\lambda q\ )q\in Q\ \}$$

Donde:

- QQ es un conjunto finito de estados.

- $q0 \in Q q0 \in Q$ es el estado inicial.
- $\Sigma\Sigma$ es un alfabeto finito.
- $\delta: Q \times \Sigma \rightarrow Q \delta: Q \times \Sigma \rightarrow Q$ es la función de transición.
- $V0 V0$ es un conjunto de valores.
- $v0 \in V0 v0 \in V0$ es el valor inicial.
- $E E$ es un conjunto finito de emisiones.
- $\mu q \in E \mu q \in E$ es la emisión asociada al estado $q q$.
- $\lambda q: V0 \times E \rightarrow V0 \lambda q : V0 \times E \rightarrow V0$ es una operación binaria asociada al estado $q q$.

B. AUTOMATAS ARITMÉTICAS PROBABILÍSTICAS

Las Automatas Aritméticas Probabilísticas (PAA) pueden definirse para formular cadenas de operaciones con operandos probabilísticos. Las PAAs pueden interpretarse como Procesos Aditivos de Markov generalizados (MAP) en casos de prueba discretos. Una Automata Aritmética Probabilística $M M$ se define como una tupla:

$M=\{Q, \delta P, V0, E, v0, q0, \mu=(\mu q) q \in Q, \lambda=(\lambda q) q \in Q\} M=\{Q, \delta P, V0, E, v0, q0, \mu=(\mu q) q \in Q, \lambda=(\lambda q) q \in Q\}$

Donde:

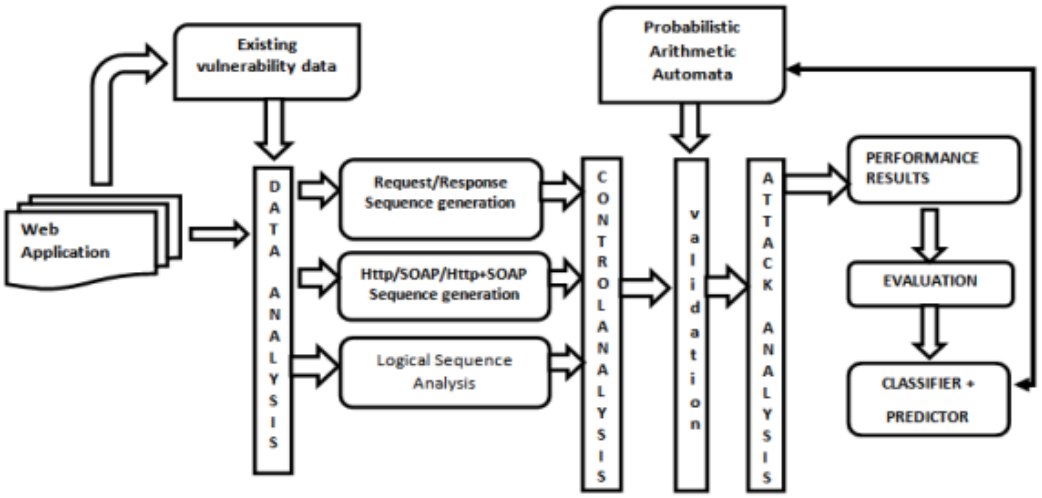
- $Q Q$ es un conjunto finito de estados.
- $q0 \in Q q0 \in Q$ es el estado inicial.
- $\delta P: Q \times Q \rightarrow [0,1] \delta P : Q \times Q \rightarrow [0,1]$ es una función de transición.
- $V0 V0$ es un conjunto llamado conjunto de valores.
- $v0 \in V v0 \in V$ es el valor inicial.
- $E E$ es un conjunto finito llamado conjunto de emisiones.

Funcionamiento de los PAA: Los PAA se utilizan para modelar operaciones probabilísticas sobre secuencias de datos. Estos modelos permiten evaluar el comportamiento de las secuencias y determinar la probabilidad de ocurrencia de ataques en base a ciertos patrones identificados durante el análisis.

4. Evaluación de Vulnerabilidades: Durante el análisis, los PAA detectan patrones de comportamiento sospechoso que pueden indicar la presencia de vulnerabilidades en las aplicaciones web. Estas vulnerabilidades se identifican mediante la evaluación de las probabilidades de emisión de ataques en cada estado del PAA.

5. Prevención de Vulnerabilidades: Además de la detección, el sistema propuesto busca prevenir la explotación de vulnerabilidades proactivamente. Esto se logra identificando y mitigando posibles puntos de entrada para ataques en las aplicaciones web.

ARQUITECTURA



TABLA

DAA Tuples	Description	Mapping DAA tuples to vulnerability assessment scenario
Q	Finite set of states	Q represents set of states i.elogin, Chk_credentials, Chk_DB, Chk_ticketStatus as shown in Figure 2.
Σ	Set of input sequences	Σ represents the input sequencesHttp Request/Response, Soap Request/Response, REST Request/Response as shown in Figure 2.
δ	Transition function	δ represents Transition from one state (login) to next state (Chk_credential) based on input sequence as shown in Figure 2.
$\mathcal{V}'$	Set of values	$\mathcal{V}'$ represent Binary operation (AND ,OR, NOT) associated to state q
E	Set of emissions	E represents total number of attacks (SQL injection, XSS attack, XML injection, replay attack) considered for vulnerability assessment

Algoritmo para el Sistema Propuesto: Detalles y Implementación

Input: pattern generated for sequences from test data

Output: probabilistic value for cause of attack in state transitions

Method:

BEGIN

//DAA Construction

Input → get input(seq)

State → q0

do

{ nextstate = STT(state, input)

state = nextstate

// Looping the number of steps using for loop. And the steps depend on number of sequence

for(each t = 1 to n)

{

//Calculate cause function Pk+1 start state (q) and value set (v) initialize as 0 for all states after k steps.

Initialize Pi+1 (q, v) = 0 for all q ∈ Q, v ∈ v 0 for all q ∈ Q and , v ∈ v 0 − 1 do

for (each transition q 0 ∈ Q and e ∈ E)

{

//Calculate emission distribution (πq) using emitted value Zt and emission set (E)

μq = Zt/E do

μ = (μq)qQ v 0 ← λq(v, e)

//Now the attack cause function is calculated using the q and v value after time steps with transition state and emission state.

Pk+1 (q, v) ← pk q 0 , v 0

.T q 0 , q .πq (e) }

}

} while (input == NULL) && (state == q0)

END //

ALGORITHM FOR REGEX PATTERN MATCHING

Input: Patrón de ataque que ocurre en las secuencias, secuencias agrupadas por servicio involucrado

Output: Distribución de emisión en los estados

BEGIN

// Inicializar variables de conteo

setemission_val to 0

setpat_occur to 0

// Definir los patrones de ataque

pat ← [pattern1, pattern2, ..., pattern n]

// Leer el archivo que contiene las secuencias

f ← read(file)

// Abrir el archivo para almacenar la distribución de emisión para cada transición

s ← open(value_file)

// El archivo value_file se utiliza para almacenar la distribución de emisión

// para cada transición, con líneas delimitadas por el símbolo '@' para distinguir las secuencias

```
// Loop a través de cada palabra en el archivo de secuencias
for word in f:
{
  // Buscar coincidencias de patrones en la palabra
  if search(pat, word):
  {
    // Si se encuentra una coincidencia, incrementar el contador de ocurrencias de patrones
    incrementar pat_occur en 1
  }
  // Verificar si hay alguna ocurrencia de patrones
  if (pat_occur > 0):
  {
    // Si hay al menos una ocurrencia de patrón, incrementar el valor de emisión
    incrementar emission_val en 1
    // Escribir el valor de emisión en el archivo de salida
    s ← write (emission_val)
  }
}
END
```

El algoritmo diseñado para el sistema propuesto desglosa el proceso de construcción de estados y cálculo de la causa de ataques en cada transición de estado. Este algoritmo es esencial para la implementación y funcionamiento efectivo del sistema de detección y

prevención de vulnerabilidades. A continuación, se presenta el algoritmo junto con una descripción detallada de su funcionamiento.

1. Construcción de Estados y Cálculo de la Causa del Ataque: El algoritmo recibe como entrada los patrones generados para las secuencias de datos provenientes de los datos de prueba. A partir de estos patrones, se construyen estados y se calcula la probabilidad de ocurrencia de ataques en cada transición de estado.

2. Método: El algoritmo sigue los siguientes pasos:

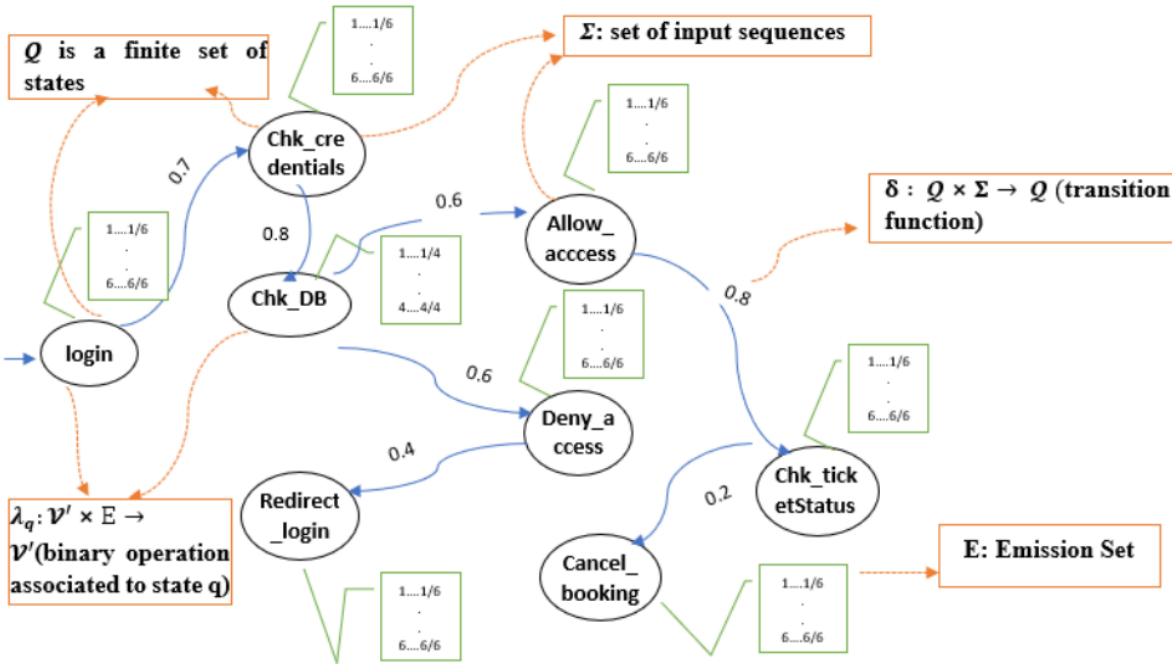
- **Construcción del DAA (Automata Aritmético Determinista):** Se construye un DAA para modelar el comportamiento determinista en la secuencia de datos. Esto se logra mediante la identificación de estados y transiciones basadas en los datos de entrada.
- **Cálculo de la Causa del Ataque:** Se calcula la probabilidad de ocurrencia de ataques en cada estado y transición. Esto se realiza mediante el cálculo de la función de causa de ataque (Pk+1) para cada estado después de un número específico de pasos.

3. Implementación del Algoritmo: El algoritmo se implementa utilizando un enfoque de programación estructurada. Se utilizan bucles y condiciones para iterar sobre los datos de entrada y calcular la probabilidad de ocurrencia de ataques en cada estado y transición.

4. Resultado: El resultado del algoritmo es una evaluación detallada de la probabilidad de ocurrencia de ataques en cada estado y transición. Esta información es fundamental para la detección y prevención efectiva de vulnerabilidades en entornos web heterogéneos.

5. Algoritmo para Coincidencia de Patrones con Expresiones Regulares: Además del algoritmo principal, se proporciona un algoritmo auxiliar para la coincidencia de patrones utilizando expresiones regulares en Python. Este algoritmo se utiliza para identificar patrones de ataques en las secuencias de datos.

AUTOMATA



CONCLUSIONES DEL AUTOR:

El modelo predictivo propuesto constituye una plataforma sólida para hacer frente a las tendencias recientes en entornos web heterogéneos. Su capacidad para identificar vulnerabilidades y ataques en este complejo entorno es destacable. La análisis persigue proporcionar un análisis profundo de las aplicaciones web, centrándose especialmente en descubrir las vulnerabilidades y su frecuencia de ocurrencia.

Para cumplir con este objetivo, el sistema toma secuencias como entrada, las cuales son analizadas en función de su frecuencia de aparición. Luego, se emplea un algoritmo de coincidencia de patrones para identificar los valores correspondientes a diferentes grupos de secuencias, así como la distribución de emisiones entre los estados de estos grupos. El resultado final es la causa general del ataque, que puede utilizarse para determinar el porcentaje de vulnerabilidad presente en la aplicación web examinada.

El modelo actual implementa eficazmente la detección de ataques, pero aún puede mejorarse mediante la reconfiguración de los Autómatas Aritméticos Probabilísticos (PAA). La salida de PAA proporciona un análisis de los ataques y su ocurrencia en cada estado.Para aumentar su poder expresivo, en el futuro se propone alimentar el informe de análisis generado por PAA a técnicas de aprendizaje automático, como cadenas de Markov o Modelos Ocultos de Markov (HMM). Este proceso podría reconfigurar el modelo probabilístico desarrollado al iterar sobre patrones recién identificados basados en estados, asegurando así no solo la detección, sino también la prevención de futuros ataques.

Además, el sistema automatizado puede implementarse en diversas aplicaciones web para garantizar su eficiencia en análisis predictivos, contribuyendo así a un entorno web más seguro en general.

CONCLUSIONES DEL ALUMNO:

La propuesta de un modelo predictivo para la detección de vulnerabilidades en entornos web heterogéneos es, sin duda, un avance significativo en la seguridad cibernética. Sin embargo, al examinar detenidamente las conclusiones presentadas por el autor, surge la necesidad de abordar ciertos aspectos críticos y considerar posibles objeciones.

En primer lugar, el enfoque en la detección y prevención de vulnerabilidades es loable, pero es crucial reconocer que ningún sistema puede garantizar una protección total contra ataques cibernéticos. Si bien el modelo propuesto puede identificar y mitigar una amplia gama de amenazas, siempre existirá la posibilidad de que nuevas vulnerabilidades escapen a su detección.

Además, la dependencia exclusiva de los Autómatas Aritméticos Probabilísticos (PAA) puede limitar la capacidad del modelo para adaptarse a la evolución constante de las tácticas de ataque. Aunque se sugiere la integración con técnicas de aprendizaje automático, como las cadenas de Markov o los Modelos Ocultos de Markov (HMM), es necesario abordar cómo estas incorporaciones pueden afectar la complejidad y la eficacia del sistema.

Otra preocupación radica en la generalización del modelo a diversas aplicaciones web. Si bien es deseable una solución universal, la diversidad de arquitecturas y tecnologías utilizadas en diferentes aplicaciones podría plantear desafíos significativos para la implementación y adaptación efectiva del modelo propuesto.

Además, se debe tener en cuenta el posible sesgo inherente en los datos de entrenamiento utilizados para desarrollar el modelo. Si los datos no representan adecuadamente la diversidad y la complejidad de los ataques potenciales, el modelo podría ser menos efectivo en la detección de amenazas reales en entornos web del mundo real.

En conclusión, si bien el modelo predictivo para la detección de vulnerabilidades en entornos web heterogéneos representa un avance prometedor, es fundamental abordar las limitaciones y desafíos identificados para garantizar su eficacia y aplicabilidad en el mundo real. Un enfoque crítico y reflexivo es esencial para avanzar hacia soluciones más sólidas y resilientes en la seguridad cibernética.

REFERENCIAS

Moshika, A., Thirumaran, M., Natarajan, B., Andal, K., Sambasivam, G., & Manoharan, R. (2021). Vulnerability Assessment in Heterogeneous Web Environment Using Probabilistic Arithmetic Automata. *IEEE Access*. https://doi.org/10.1109/ACCESS.2021.3081567