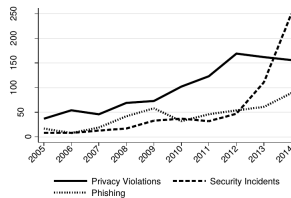
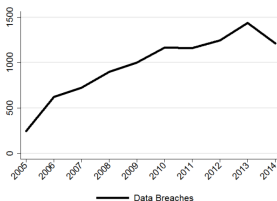


Viterbi Algorithm for Intrusion Type Identification in Anomaly Detection System

january 14th 2019

Context

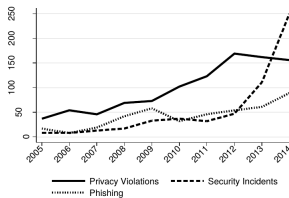
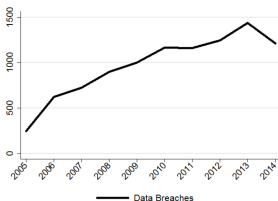


1

- Number of intrusions is increasing with time and can cause a lot of damage

¹Sasha Romanosky (2016). "Examining the costs and causes of cyber incidents". In: *Journal of Cybersecurity* 2.2, pp. 121–135

Context



1

- Number of intrusions is increasing with time and can cause a lot of damage
- In 2005, among 7,818 businesses
 - Nearly 60% detected one or more types of cyber attack. (*National Computer Security Survey (NCSS)*)
 - Approximately 68% of the victims of cyber theft sustained monetary loss of \$10,000 or more.

¹Sasha Romanosky (2016). "Examining the costs and causes of cyber incidents". In: *Journal of Cybersecurity* 2.2, pp. 121–135

Intrusion Type

- . Buffer overflow
 - . xlock vulnerability
 - . lpset vulnerability
 - . kcms_sparc vulnerability
- . S/W security vulnerability
- . Setup vulnerability
- . Denial of service

Intrusion Detection Systems (IDS)

- . **host-based**: related to OS information
- . **network based**: network related events
- . **misuse-based**: seek defined patterns, or signatures, within the analyzed data
- . **anomaly-based**: estimate the “normal” behaviour of the system to be protected, and generate an anomaly alarm whenever the deviation between a given observation at an instant and the normal behaviour exceeds a predefined threshold

Intrusion Detection Systems (IDS)

- . **host-based**: related to OS information
- . **network based**: network related events
- . **misuse-based**: seek defined patterns, or signatures, within the analyzed data
- . **anomaly-based**: estimate the “normal” behaviour of the system to be protected, and generate an anomaly alarm whenever the deviation between a given observation at an instant and the normal behaviour exceeds a predefined threshold

HMM - Hidden Markov Model

- Hidden Markov Model is a statistical model in which the modeled system is supposed to be a Markovian process of unknown parameters.

HMM - Hidden Markov Model

- Hidden Markov Model is a statistical model in which the modeled system is supposed to be a Markovian process of unknown parameters.
- Hidden Markov Model can be viewed as a Bayesian Network

HMM - Hidden Markov Model

- Hidden Markov Model is a statistical model in which the modeled system is supposed to be a Markovian process of unknown parameters.
- Hidden Markov Model can be viewed as a Bayesian Network
- We define a HMM including :
 - V, A finite set of M observations
 - B, A a matrix of probabilities of observations over state :

$$b_i(k) = P(o_t = V_k | S_t = i)$$

HMM - Forward Algorithm

input : λ The model, O Observed sequence

output : $P(O|\lambda)$

Step 1, Initialization : $\forall i, \alpha_1(i) = \pi_i b_i(O_1)$

Step 2, Induction :

for $t \leftarrow 2 : T$ **do**

$$\left| \quad \forall i, \alpha_t(i) = \left[\sum_{j=1}^N \alpha_{t-1}(j) a_{ij} \right] b_j(O_t) \right.$$

2

end

Step 3, Termination : $P(O|\lambda) = \sum_{i=1}^N \alpha_t(i)$

²L. R. Rabiner (1989). "A tutorial on hidden Markov models and selected applications in speech recognition". In: *Proceedings of the IEEE* 77.2, pp. 257–286

HMM - Viterbi Algorithm

input : O Observed sequence

output : $\arg \max_{\lambda \in \Lambda} P(0|\lambda)$

Step 1, Initialization :

for $i \leftarrow 1 : N$ **do**
$$| \quad \delta_1(i) = \pi_i b_i(0_1)$$
$$\psi_1(i) = 0$$

end

Step 2, Recursion :

for $t \leftarrow 2 : T$ do**for** $j \leftarrow 1 : N$ **do**
$$\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$$
$$\psi_t(j) = \arg \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$$

end

end

Step 3. Termination :

$$P^* = \max_{s \in S} [\delta_T(s)]$$
$$S_T^* = \arg \max_{s \in S} [\delta_T(s)]$$

Step 4, Backtracking :

```

for  $t \leftarrow T - 1 : 1$  do

```

$$S_t^* = \psi_{t+1}(s_{t+1}^*)$$

end

return S^*

³A. Viterbi (1967). "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". In: *IEEE Transactions on Information Theory* 13.2, pp. 260–269

Normal Behaviour Modeling

Normal Behaviour is modelised by a left-to-right HMM λ .

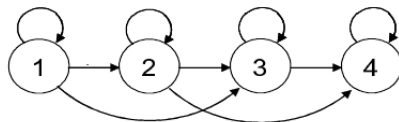


Figure: Left-to-Right Model with jumps

The forward algorithm is used to decide whether normal or not with a threshold.

Intrusion Detection

Data

$$S = \{1, 2, 3, 4\}$$

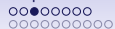
$$M = \{1, 2, 3, 4\}$$

$$\pi = \{1.0, 0, 0\}$$

$$O = \{2, 1, 2, 4, 2, 3, 4, 3, 4\}$$

$$A = \begin{pmatrix} 0.28 & 0.34 & 0.28 & 0 \\ 0.0 & 0.32 & 0.21 & 0.47 \\ 0.0 & 0.0 & 0.32 & 0.68 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0.8 & 0.04 & 0.1 & 0.06 \\ 0.0 & 0.13 & 0.45 & 0.42 \\ 0.0 & 0.9 & 0.1 & 0.0 \\ 0.64 & 0.12 & 0.06 & 0.18 \end{pmatrix}$$



Intrusion Detection

Initialization

$$\forall i, \alpha_1(i) = \pi_i b_i(0_1)$$



Intrusion Detection

Initialization

$$\forall i, \alpha_1(i) = \pi_i b_i(0_1)$$

$$\begin{aligned} O_1 &= 2 \\ b_i(0_1) &= (0.04, 0.13, 0.9, 0.12) \end{aligned}$$

Intrusion Detection

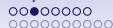
Initialization

$$\forall i, \alpha_1(i) = \pi_i b_i(0_1)$$

$$O_1 = 2$$

$$b_i(0_1) = (0.04, 0.13, 0.9, 0.12)$$

$$\alpha_1(1) = \pi_1 * b_1(0_1) = 1 * 0.04 = 0.04$$



Intrusion Detection

Initialization

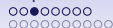
$$\forall i, \alpha_1(i) = \pi_i b_i(0_1)$$

$$O_1 = 2$$

$$b_i(0_1) = (0.04, 0.13, 0.9, 0.12)$$

$$\alpha_1(1) = \pi_1 * b_1(0_1) = 1 * 0.04 = 0.04$$

$$\alpha_1(2) = \pi_2 * b_2(0_1) = 0 * 0.13 = 0$$



Intrusion Detection

Initialization

$$\forall i, \alpha_1(i) = \pi_i b_i(0_1)$$

$$O_1 = 2$$

$$b_i(0_1) = (0.04, 0.13, 0.9, 0.12)$$

$$\alpha_1(1) = \pi_1 * b_1(0_1) = 1 * 0.04 = 0.04$$

$$\alpha_1(2) = \pi_2 * b_2(0_1) = 0 * 0.13 = 0$$

...

$$\alpha_1 = \begin{pmatrix} 0.04 & 0 & 0 & 0 \end{pmatrix}$$



Intrusion Detection

Induction

for $t \leftarrow 2 : T$ **do**

$$\left| \forall i, \alpha_t(i) = \left[\sum_{j=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(O_t) \right.$$

end

Intrusion Detection

Induction

for $t \leftarrow 2 : T$ **do**

$$\left| \quad \forall i, \alpha_t(i) = \left[\sum_{j=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(O_t) \right.$$

end

$$t = 2$$

$$O_2 = 1$$

$$b(O_t) = \begin{pmatrix} 0.8 & 0 & 0 & 0.64 \end{pmatrix}$$

$$\alpha_1 = \begin{pmatrix} 0.04 & 0 & 0 & 0 \end{pmatrix}$$

Intrusion Detection

Induction

for $t \leftarrow 2 : T$ **do**

$$\quad \left| \quad \forall i, \alpha_t(i) = \left[\sum_{j=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(O_t)$$

end

$$t = 2$$

$$O_2 = 1$$

$$b(O_t) = \begin{pmatrix} 0.8 & 0 & 0 & 0.64 \end{pmatrix}$$

$$\alpha_1 = \begin{pmatrix} 0.04 & 0 & 0 & 0 \end{pmatrix}$$

$$\alpha_2(1) = \left[\sum_{j=1}^N \alpha_{t-1}(1) a_{1j} \right] b_j(O_t) = 0.00896$$



Intrusion Detection

Induction

for $t \leftarrow 2 : T$ **do**

$$\quad \left| \quad \forall i, \alpha_t(i) = \left[\sum_{j=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(O_t)$$

end

$$t = 2$$

$$O_2 = 1$$

$$b(O_t) = \begin{pmatrix} 0.8 & 0 & 0 & 0.64 \end{pmatrix}$$

$$\alpha_1 = \begin{pmatrix} 0.04 & 0 & 0 & 0 \end{pmatrix}$$

$$\alpha_2(1) = \left[\sum_{j=1}^N \alpha_{t-1}(1) a_{1j} \right] b_j(O_t) = 0.00896$$

...

$$\alpha_2 = \begin{pmatrix} 0.00896 & 0 & 0 & 0 \end{pmatrix}$$



Intrusion Detection

Induction

for $t \leftarrow 2 : T$ **do**

$$\left| \begin{array}{l} \forall i, \alpha_t(i) = \left[\sum_{j=1}^N \alpha_{t-1}(i) a_{ij} \right] b_j(O_t) \end{array} \right.$$

end

$$\alpha = \begin{pmatrix} 0.04 & 0 & 0 & 0 \\ 0.00896 & 0 & 0 & 0 \\ 0.00010035 & 0.00039603 & 0.0022579 & 0 \\ 1.8882e^{-08} & 2.8849e^{-06} & 1.3193e^{-05} & 4.0995e^{-05} \\ 5.287e^{-10} & 4.1831e^{-07} & 4.8329e^{-07} & 3.0793e^{-06} \\ 8.8822e^{-12} & 5.6297e^{-08} & 0 & 6.4882e^{-07} \\ 2.487e^{-13} & 8.1081e^{-09} & 1.1825e^{-09} & 4.0517e^{-08} \\ 4.1782e^{-15} & 1.0898e^{-09} & 0 & 8.1237e^{-09} \\ 1.1699e^{-16} & 1.5693e^{-10} & 2.2885e^{-11} & 5.1816e^{-10} \end{pmatrix}$$



Intrusion Detection

Termination

$$\begin{aligned}
 P(0|\lambda)) &= \sum_{i=1}^N \alpha_t(i) \\
 &= 1.1699e^{-16} + 1.5693e^{-10} + 2.2885e^{-11} + 5.1816e^{-10} \\
 &= 6.9797e^{-10}
 \end{aligned}$$



Intrusion Detection

Decision

```
if  $\log(P(0|\lambda)) > \text{threshold}$  then  
  | return Normal Behaviour  
else  
  | return Intrusion  
end
```

$$\log(P(0|\lambda)) = -21.083 < \text{threshold}(-20.83) \implies \text{Intrusion}$$

Intrusion Detection

Results

Table: The performance of HMM-based IDS. Best results are in bold

Length	Thresold	Detection Rate	F-P Error
10	-9.43	100%	2.626
15	-9.43	100%	3.614
10	-14.42	100%	1.366
15	-14.42	100%	2.718
10	-16.94	100%	0.789
15	-16.94	100%	2.618
10	-18.35	100%	0.553
15	-18.35	100%	2.535
10	-19.63	100%	0.476
15	-19.63	100%	2.508
10	-20.83	100%	0.372
15	-20.83	100%	2.473



Intrusion Type Identification

Process in two steps :

- Viterbi algorithm used to find the optimal state sequence
- Euclidean distance to identify the intrusion type with the optimal state sequence

Intrusion Type Identification

Data

$$S = \{1, 2, 3, 4\}$$

$$M = \{1, 2, 3, 4\}$$

$$\pi = \{1.0, 0, 0\}$$

$$O = \{2, 1, 2, 4, 2, 3, 4, 3, 4, 3\}$$

$$A = \begin{pmatrix} 0.28 & 0.34 & 0.28 & 0 \\ 0.0 & 0.32 & 0.21 & 0.47 \\ 0.0 & 0.0 & 0.32 & 0.68 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0.8 & 0.04 & 0.1 & 0.06 \\ 0.0 & 0.13 & 0.45 & 0.42 \\ 0.0 & 0.9 & 0.1 & 0.0 \\ 0.64 & 0.12 & 0.06 & 0.18 \end{pmatrix}$$

Intrusion Type Identification

Initialization

```
for  $i \leftarrow 1 : N$  do
  |  $\delta_1(i) = \pi_i b_i(0_1)$ 
  |  $\psi_1(i) = 0$ 
end
```

$$O_1 = 2$$

$$b_i(0_1) = (0.04, 0.13, 0.9, 0.12)$$



Intrusion Type Identification

Initialization

```
for  $i \leftarrow 1 : N$  do  
   $\delta_1(i) = \pi_i b_i(0_1)$   
   $\psi_1(i) = 0$   
end
```

$$\begin{aligned} O_1 &= 2 \\ b_i(0_1) &= (0.04, 0.13, 0.9, 0.12) \\ \delta_1(1) &= \pi_1 * b_1(0_1) = 1 * 0.04 = 0.04 \end{aligned}$$

Intrusion Type Identification

Initialization

```
for  $i \leftarrow 1 : N$  do
  |  $\delta_1(i) = \pi_i b_i(0_1)$ 
  |  $\psi_1(i) = 0$ 
end
```

$O_1 =$	2
$b_i(0_1) =$	(0.04, 0.13, 0.9, 0.12)
$\delta_1(1) =$	$\pi_1 * b_1(0_1) = 1 * 0.04 = 0.04$
$\delta_1(2) =$	$\pi_2 * b_2(0_1) = 0 * 0.13 = 0$

Intrusion Type Identification

Initialization

```

for  $i \leftarrow 1 : N$  do
  |  $\delta_1(i) = \pi_i b_i(0_1)$ 
  |  $\psi_1(i) = 0$ 
end

```

$$\begin{aligned}
 O_1 &= 2 \\
 b_i(0_1) &= (0.04, 0.13, 0.9, 0.12) \\
 \delta_1(1) &= \pi_1 * b_1(0_1) = 1 * 0.04 = 0.04 \\
 \delta_1(2) &= \pi_2 * b_2(0_1) = 0 * 0.13 = 0 \\
 &\dots \\
 \delta_1 &= (0.04 \quad 0 \quad 0 \quad 0)
 \end{aligned}$$



Intrusion Type Identification

Initialization

```
for  $i \leftarrow 1 : N$  do
  |  $\delta_1(i) = \pi_i b_i(0_1)$ 
  |  $\psi_1(i) = 0$ 
end
```

$$\begin{aligned}
 O_1 &= 2 \\
 b_i(0_1) &= (0.04, 0.13, 0.9, 0.12) \\
 \delta_1(1) &= \pi_1 * b_1(0_1) = 1 * 0.04 = 0.04 \\
 \delta_1(2) &= \pi_2 * b_2(0_1) = 0 * 0.13 = 0 \\
 &\dots \\
 \delta_1 &= \begin{pmatrix} 0.04 & 0 & 0 & 0 \end{pmatrix}
 \end{aligned}$$

$$\psi_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}$$



Intrusion Type Identification

Recursion

```

for  $t \leftarrow 2 : T$  do
  | for  $j \leftarrow 1 : N$  do
  | |  $\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$ 
  | |  $\psi_t(j) = \arg \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$ 
  | end
end

```

$$t = 2$$

$$O_2 = 1$$

$$\delta_1 = \begin{pmatrix} 0.04 & 0 & 0 & 0 \end{pmatrix}$$



Intrusion Type Identification

Recursion

```

for  $t \leftarrow 2 : T$  do
  for  $j \leftarrow 1 : N$  do
     $\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$ 
     $\psi_t(j) = \arg \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$ 
  end
end

```

$$\begin{aligned}
 t &= 2 \\
 O_2 &= 1 \\
 \delta_1 &= \begin{pmatrix} 0.04 & 0 & 0 & 0 \end{pmatrix} \\
 \delta_2(1) &= \max_i [\delta_{t-1}(i) a_{i1}] b_1(0_2) \\
 &= 0.00896 \\
 \delta_2 &= \begin{pmatrix} 0.00896 & 0 & 0 & 0 \end{pmatrix}
 \end{aligned}$$



Intrusion Type Identification

Recursion

```

for  $t \leftarrow 2 : T$  do
  for  $j \leftarrow 1 : N$  do
     $\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$ 
     $\psi_t(j) = \arg \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$ 
  end
end

```

$$\begin{aligned}
 t &= 2 \\
 O_2 &= 1 \\
 \delta_1 &= \begin{pmatrix} 0.04 & 0 & 0 & 0 \end{pmatrix} \\
 \delta_2(1) &= \max_i [\delta_{t-1}(i) a_{i1}] b_1(0_2) \\
 &= 0.00896 \\
 \delta_2 &= \begin{pmatrix} 0.00896 & 0 & 0 & 0 \end{pmatrix} \\
 \psi_2(1) &= \arg \max_i [\delta_{t-1}(i) a_{i1}] b_1(0_2) \\
 &= 0
 \end{aligned}$$



Intrusion Type Identification

Recursion

```

for  $t \leftarrow 2 : T$  do
  for  $j \leftarrow 1 : N$  do
     $\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$ 
     $\psi_t(j) = \arg \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$ 
  end
end

```

$$\begin{aligned}
 t &= 2 \\
 O_2 &= 1 \\
 \delta_1 &= \begin{pmatrix} 0.04 & 0 & 0 & 0 \end{pmatrix} \\
 \delta_2(1) &= \max_i [\delta_{t-1}(i) a_{i1}] b_1(0_2) \\
 &= 0.00896 \\
 \delta_2 &= \begin{pmatrix} 0.00896 & 0 & 0 & 0 \end{pmatrix} \\
 \psi_2(1) &= \arg \max_i [\delta_{t-1}(i) a_{i1}] b_1(0_2) \\
 &= 0 \\
 \psi_2 &= \begin{pmatrix} 0 & 0 & 0 & 0 \end{pmatrix}
 \end{aligned}$$

Intrusion Type Identification

Recursion

```

for  $t \leftarrow 2 : T$  do
  for  $j \leftarrow 1 : N$  do
     $\delta_t(j) = \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$ 
     $\psi_t(j) = \arg \max_i [\delta_{t-1}(i) a_{ij}] b_j(0_t)$ 
  end
end
end

```

$$\delta = \begin{pmatrix} 0.04 & 0 & 0 & 0 \\ 0.00896 & 0 & 0 & 0 \\ 0.00010035 & 0.00039603 & 0.0022579 & 0 \\ 1.6859e^{-06} & 5.3227e^{-05} & 0 & 0.00027637 \\ 1.8882e^{-08} & 2.2142e^{-06} & 1.006e^{-05} & 3.3164e^{-05} \\ 5.287e^{-10} & 3.1885e^{-07} & 3.2192e^{-07} & 1.9899e^{-06} \\ 8.8822e^{-12} & 4.2853e^{-08} & 0 & 3.5817e^{-07} \\ 2.487e^{-13} & 6.1709e^{-09} & 8.9992e^{-10} & 2.149e^{-08} \\ 4.1782e^{-15} & 8.2937e^{-10} & 0 & 3.8683e^{-09} \\ 1.1699e^{-16} & 1.1943e^{-10} & 1.7417e^{-11} & 2.321e^{-10} \end{pmatrix}$$

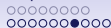
$$\psi = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 1 & 1 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 0 & 3 \\ 0 & 1 & 1 & 3 \\ 0 & 1 & 0 & 3 \\ 0 & 1 & 1 & 3 \end{pmatrix}$$



Intrusion Type Identification

Termination

$$P^* = \max_{s \in S} [\delta_T(s)] = 2.321e^{-10}$$



Intrusion Type Identification

Backtracking

```
for  $t \leftarrow T - 1 : 1$  do  
   $S_t^* = \psi_{t+1}(s_{t+1}^*)$   
end
```

Optimal Sequence $S^* = \{1, 1, 3, 4, 4, 4, 4, 4, 4\}$

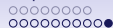


Intrusion Type Identification

Decision

Table: Sequences for each type of intrusion

Type	Sequence	Distance
xlock	{2, 2, 3, 3, 3, 4, 4, 4, 4, 4}	3.7417
ipset	{2, 3, 3, 3, 4, 4, 4, 4, 4, 4}	4.4721
kcms_sparc	{1, 1, 2, 2, 2, 2, 4, 4, 4, 4}	3



Intrusion Type Identification

Results

Table: The performance of Viterbi-based Intrusion Type Identification

Attack	Trial	Correct	Incorrect	Rate
Buffer Overflow	20	18	2	90%
Denial of Service	25	9	16	36%
All	45	27	18	60%

Limitations & Remarks

- Try other distance metrics for Intrusion Type Identification :
Ja-Min Koo and Sung-Bae Cho (2005). “Effective Intrusion Type Identification with Edit Distance for HMM-Based Anomaly Detection System”. In: *Pattern Recognition and Machine Intelligence*. Ed. by Sankar K. Pal, Sanghamitra Bandyopadhyay, and Sambhunath Biswas. Springer Berlin Heidelberg

Limitations & Remarks

- Try other distance metrics for Intrusion Type Identification :
Ja-Min Koo and Sung-Bae Cho (2005). “Effective Intrusion Type Identification with Edit Distance for HMM-Based Anomaly Detection System”. In: *Pattern Recognition and Machine Intelligence*. Ed. by Sankar K. Pal, Sanghamitra Bandyopadhyay, and Sambhunath Biswas. Springer Berlin Heidelberg
- Hypothesis that there is only one sequence of state per each intrusion, and that it never changes.

Limitations & Remarks

- Try other distance metrics for Intrusion Type Identification :
[Ja-Min Koo and Sung-Bae Cho \(2005\)](#). “Effective Intrusion Type Identification with Edit Distance for HMM-Based Anomaly Detection System”. In: *Pattern Recognition and Machine Intelligence*. Ed. by Sankar K. Pal, Sanghamitra Bandyopadhyay, and Sambhunath Biswas. Springer Berlin Heidelberg
- Hypothesis that there is only one sequence of state per each intrusion, and that it never changes.
- This model is anomaly-based, but use the fact that we are supposed to know the sequence of state of the intrusion. they loose the main advantage of anomaly-based IDS to detect new types of intrusion.

Limitations & Remarks

- Low detection efficiency, especially due to the high false positive rate usually obtained [Stefan Axelsson \(1998\)](#).
Research in intrusion-detection systems: A survey. Tech. rep. Technical report 98–17. Department of Computer Engineering, Chalmers ...

Limitations & Remarks

- Low detection efficiency, especially due to the high false positive rate usually obtained [Stefan Axelsson \(1998\)](#).
Research in intrusion-detection systems: A survey. Tech. rep. Technical report 98–17. Department of Computer Engineering, Chalmers ...
- Absence of appropriate metrics and assessment methodologies, as well as a general framework for evaluating and comparing alternative IDS techniques [Salvatore J Stolfo et al. \(2000\)](#).
Cost-based modeling for fraud and intrusion detection: Results from the JAM project. Tech. rep. COLUMBIA UNIV NEW YORK DEPT OF COMPUTER SCIENCE

Methods using HMM

Intrusion Alert Prediction Using a Hidden Markov Mode

Alert prediction method based on prediction of the next alert cluster

Clusters contains :

- source IP address
- destination IP range
- alert type
- alert category.

Prediction of next alert cluster provides more information about future strategies of the attacker and does not depend on specific domain knowledge

4

⁴Udaya Sampath K Thanthrige, Jagath Samarabandu, and Xianbin Wang (2016). "Intrusion alert prediction using a hidden Markov model". In: *arXiv preprint arXiv:1610.07276*

Methods using HMM

Anomalybased HMMs

Used for intrusion detection, with five states and six observation symbols per state

States in the model are interconnected in such a way that any state can be reached from any other state

Baum-Welch method is used

5

⁵Shrijit S Joshi and Vir V Phoha (2005). "Investigating hidden Markov models capabilities in anomaly detection". In: *Proceedings of the 43rd annual Southeast regional conference-Volume 1*. ACM, pp. 98-103



Other Methods

Technique: basics	■ Pros	Subtypes
	■ Cons	
A) Statistical-based: <i>stochastic behaviour</i>	<ul style="list-style-type: none"> ■ Prior knowledge about normal activity not required. Accurate notification of malicious activities. ■ Susceptible to be trained by attackers. Difficult setting for parameters and metrics. Unrealistic quasi-stationary process assumption. 	<p>A.1) Univariate models (<i>independent Gaussian random variables</i>)</p> <p>A.2) Multivariate models (<i>correlations among several metrics</i>)</p> <p>A.3) Time series (<i>interval timers, counters and some other time-related metrics</i>)</p>
B) Knowledge-based: <i>availability of prior knowledge/data</i>	<ul style="list-style-type: none"> ■ Robustness. Flexibility and scalability. ■ Difficult and time-consuming availability for high-quality knowledge/data. 	<p>B.1) Finite state machines (<i>states and transitions</i>)</p> <p>B.2) Description languages (<i>N-grams, UML, ...</i>)</p> <p>B.3) Expert systems (<i>rules-based classification</i>)</p>
C) Machine learning-based: <i>categorization of patterns</i>	<ul style="list-style-type: none"> ■ Flexibility and adaptability. Capture of interdependencies. ■ High dependency on the assumption about the behaviour accepted for the system. High resource consuming. 	<p>C.1) Bayesian networks (<i>probabilistic relationships among variables</i>)</p> <p>C.2) Markov models (<i>stochastic Markov theory</i>)</p> <p>C.3) Neural networks (<i>human brain foundations</i>)</p> <p>C.4) Fuzzy logic (<i>approximation and uncertainty</i>)</p> <p>C.5) Genetic algorithms (<i>evolutionary biology inspired</i>)</p> <p>C.6) Clustering and outlier detection (<i>data grouping</i>)</p>

6

⁶Pedro Garcia-Teodoro et al. (2009). “Anomaly-based network intrusion detection: Techniques, systems and challenges”. In: *computers & security* 28.1-2, pp. 18–28

○○○○○○○○
○○○○○○○○○○

Other Methods

TABLE VII
COMPLEXITY OF ML AND DM ALGORITHMS DURING TRAINING

Algorithm	Typical Time Complexity	Streaming Capable	Comments
ANN	$O(emnk)$	low	Jain et al. [107] e: number of epochs k: number of neurons
Association Rules	$\gg O(n^3)$	low	Agrawal et al. [108]
Bayesian Network	$\gg O(mn)$	high	Jensen [41]
Clustering, k-means	$O(kmni)$	high	Jain and Dubes [46] i: number of iterations until threshold is reached k: number of clusters
Clustering, hierarchical	$O(n^3)$	low	Jain and Dubes [46]
Clustering, DBSCAN	$O(n \log n)$	high	Ester et al. [109]
Decision Trees	$O(mn^2)$	medium	Quinlan [54]
GA	$O(gkmn)$	medium	Oliveto et al. [110] g: number of generations k: population size
Naïve Bayes	$O(mn)$	high	Witten and Frank [89]
Nearest Neighbor k-NN	$O(n \log k)$	high	Witten and Frank [89] k: number of neighbors
HMM	$O(nc^2)$	medium	Forney [111] c: number of states (categories)
Random Forest	$O(Mmn \log n)$	medium	Witten and Frank [89] M: number of trees
Sequence Mining	$\gg O(n^3)$	low	Agrawal and Srikant [92]
SVMs	$O(n^3)$	medium	Burges [112]

7

⁷Anna L Buczak and Erhan Guven (2016). "A survey of data mining and machine learning methods for cyber security intrusion detection". In: *IEEE Communications Surveys & Tutorials* 18.2, pp. 1153–1176

Conclusion

Good results for Intrusion detection

For type identification :

Good results for Buffer Overflow (90%)

Bad results for Denial of Service (36%)

Any question ?