Intellectual Intrusion Detection with Sequences Alignment Method

Maxim Kalinin, Yaroslav Markov

MMM-ACNS-2010

Sequences alignment algorithms introduction

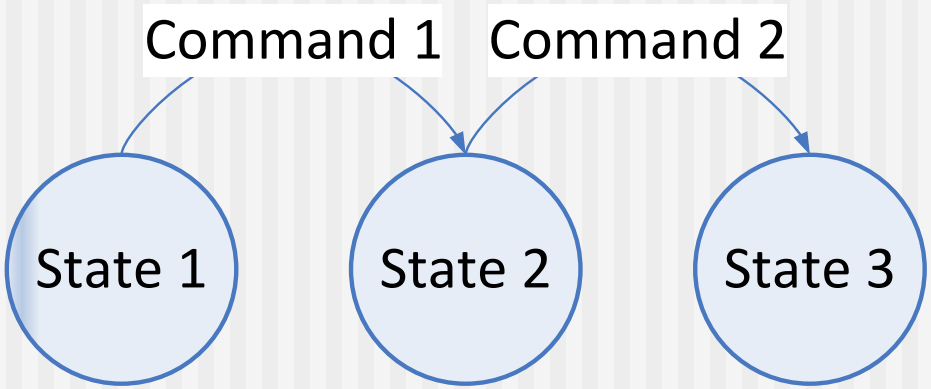
Sequences alignment algorithms (SAA) are intended to detect regions of similarity between biological sequences.

Generally they fall into two categories:

- Local alignment. (Identification of Common Molecular Subsequences. [Smith, T.F., Waterman M.S.] 1981)
- Global alignment. (A general method applicable to the search for similarities in the amino acid sequence of two proteins. [Needleman, S.B., Wunsch, C.D.] 1970)

Why SAA were chosen?

Usually Intrusion Detection System solves the problem of matching two sequences.



The following approaches can be used:

- Signature-based
- Anomaly-based

Using SAA in attack detection

- An Algorithm for Generation of Attack Signatures Based on Sequences Alignment. [Li, N., Xia, C., Yang, Y., Wang, H.] 2008
- Intrusion Detection: A Bioinformatics Approach. [Coull, S.E., Branch, J.W., Szymanski, B.K., Breimer, E.] 2003
- Network Intrusion Detection Using Genetic Algorithm to find Best DNA Signature. [Al-Ibisi T., El-Latif Abu-Dalhoum, A., Al-Rawi, M., Alfonseca M., Ortega A.] 2008

Chosen approaches: Mimicry attacks

Mimicry attacks on host-based intrusion detection systems [Wagner, D., Soto, P.] 2002

Mimicry attack is denoted as attack which allows a sophisticated attacker to cloak their intrusion to avoid detection by the IDS. These attacks are like mutations in terms of genetics.

Some ways to create mimicry attack:

- Insert no-ops
- Replace system calls parameters

Chosen approaches: Mimicry attacks

Example:

Original attack: setreuid(0,0); open("/etc/passwd", O_APPEND|O_WRONLY); write(fd, "newuser:password:0:0::/:/bin/sh", 31); close(fd); exit(0);

Mimicry attack: setreuid(0,0); brk(); brk(); brk(); setreuid(); open("/etc/passwd", O_APPEND|O_WRONLY); time(); brk(); write(fd, "newuser:password:0:0::/:/bin/sh", 31); time(); fork(); close(fd); time(); brk(); exit(0);

Chosen approaches: Mimicry attacks

Wagner and Soto used pattern matching to detect presence of mimicry attacks with no-ops.

With SAA we will try to:

- Improve efficacy
- Decrease pattern database size

Chosen approaches: Sliding Window

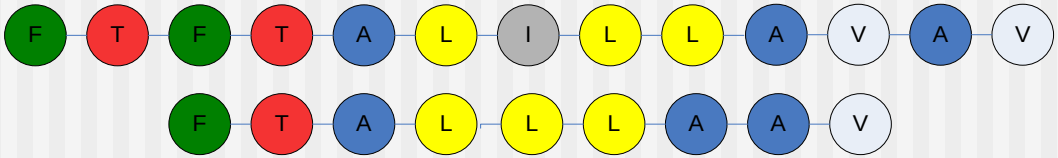
A sense of self for unix processes [Forrest, S., Hofmeyr, S.A., Somayaji, A., Longstaff, T.A.] 1996

This is anomaly-based method. It is about building database of normal behaviour on learning phase. And then comparison of analysing behaviour to built database.

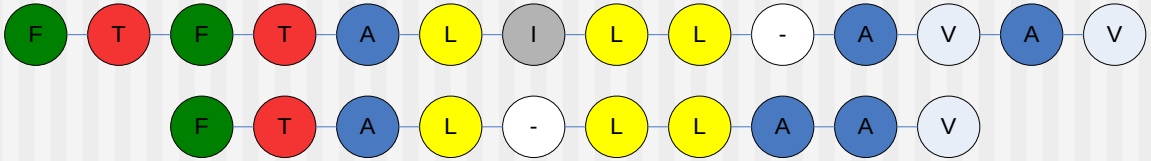
With SAA we will try to decrease database size.

Sequences Alignment Algorithms

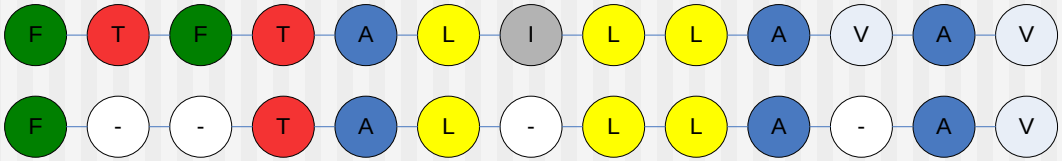
Input:



Local:



Global:



Sequences Alignment Algorithms

- Input: two sequences, function that determines similarity degree between two genes
- Output: two aligned sequences, value that determines similarity degree between sequences

Sequences alignment algorithms can be applied not only to sequences of genes, but to sequences of anything.

Local Sequences Alignment

- Two sequences: $a = «C_{a1}, C_{a2} \ldots C_{am}»,$
  $b = «C_{b1}, C_{b2} \ldots C_{bn}»$
- Similarity function: $\omega : (C_{ai} \cup -, C_{bj} \cup -,) \rightarrow Z$

Local Sequences Alignment

$$H(i,0) = 0,\ 0 \le i < m$$

$$H(0,j) = 0, 0 \le j < n$$

$$H(i,j) = max\begin{cases} 0 \\ H(i-1,j-1) + \omega(C_{ai}, C_{bi}) \\ H(i-1,j) + \omega(C_{ai}, -) \\ H(i,j-1) + \omega(-, C_{bi}) \end{cases}$$

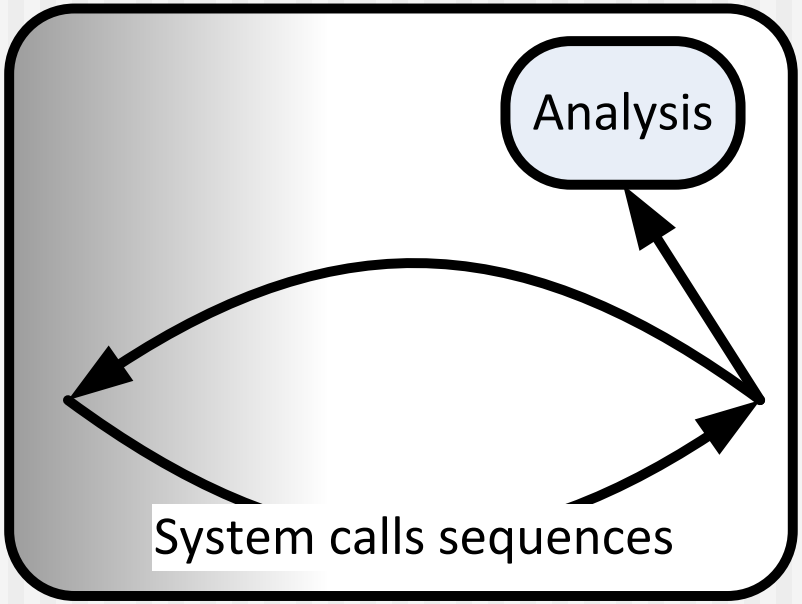
$$1 \le i \le m$$

$$1 \le j \le n$$

Local Sequences Alignment

- With the help of this matrix the sequences are aligned.
- Similarity value: $$R(a,b) = \sum_{i=0}^{n-1} \omega(C_{ai}, C_{bi})$$
- Global alignment is very similar to local alignment

What sequences will be analysed?



Attacker's
Target

Attacks that will be used

We chose datasets produced by University of New Mexico.

- Xlock – 3 buffer overflow attacks
- Sendmail – sunsendmailcp, decode, syslog-remote, syslog-local
- Lpr – lprcp

Applying SAA to mimicry attacks detection

- Let assume we have some attack signature and we want to find in given sequence S = «C_1, C_2, C_3…C_N» the presence of this attack mimicry forms.
- *S* is divided into subsequences: «C_1, C_2, C_3…C_P», «C_1, C_2, C_3…C_{P+1}» … «C_{N-P+1}, C_{N-P+2}, C_{N-P+3}…C_N». Attack signature is compared to these subsequences with use of SAA. On each comparison we get R – value of similarity between sequences.

Applying SAA to mimicry attacks detection

- *R* is normalized and compared to threshold. If R is greater than threshold, then the mimicry attack is found.

- We checked that our method can detect mimicry attacks with no-ops. Method described by Wagner and Soto is almost a particular case of sequence alignment.
- In theory our method can detect mimicry attacks that are done with adding operations that neutralize each other.
- SAA allows to define similarity function in a way that presence of some dangerous system calls will be more important that presence of others.

Sliding window

Learning phase:

- $SystemTrace$ - trace corresponding to system normal behaviour: $«C_1,C_2 \dots C_n»$
- $SystemTrace$ is divided into subsequences: $Seq = \{Seq_i, 0 \le i \le N-P+1, P \le N\}$, $Seq_i = «C_i,C_{i+1} \dots C_{P+i}»$
- $NormDb$ – subset of Seq that denotes the database of normal behaviour. Seq_i is added to this database if the following condition is true:

$$\forall Seq_j \in NormDb \wedge Seq_j \neq Seq_i$$

Sliding window

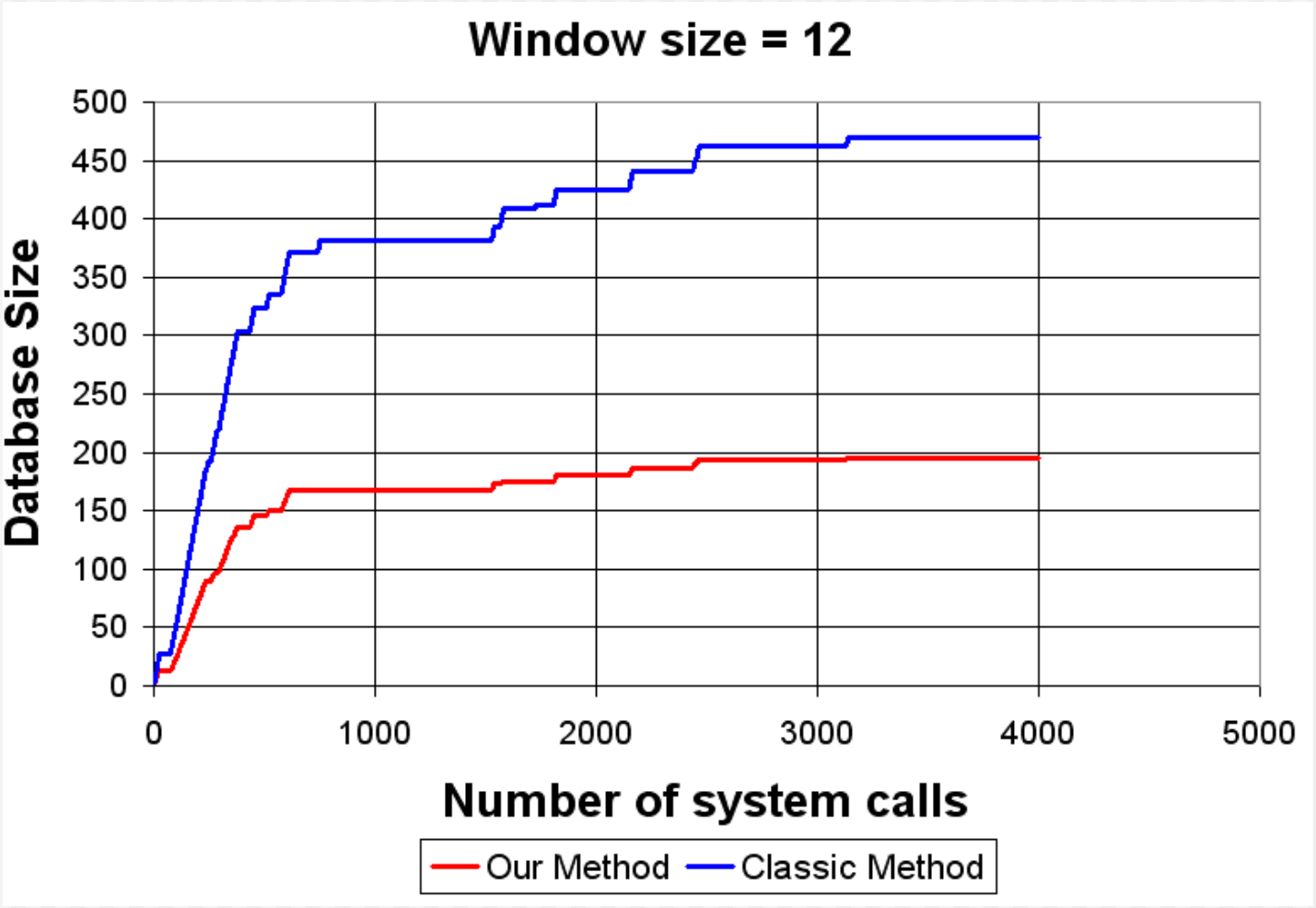
Exploitation phase:

- Trace of analysing behaviour is divided into subsequences in a manner described above: $Seq = \{Seq_i,\ 0 \le i \le N-P+1,\ P \le N\}$, $Seq_i = «C_i, C_{i+1} \dots C_{P+i}»$
- If any Seq_i is not equal to any member of $NormDb$ then it is abnormal sequence
- If percentage of abnormal sequences is greater than threshold, then it is anomalous behaviour and attack alarm is given

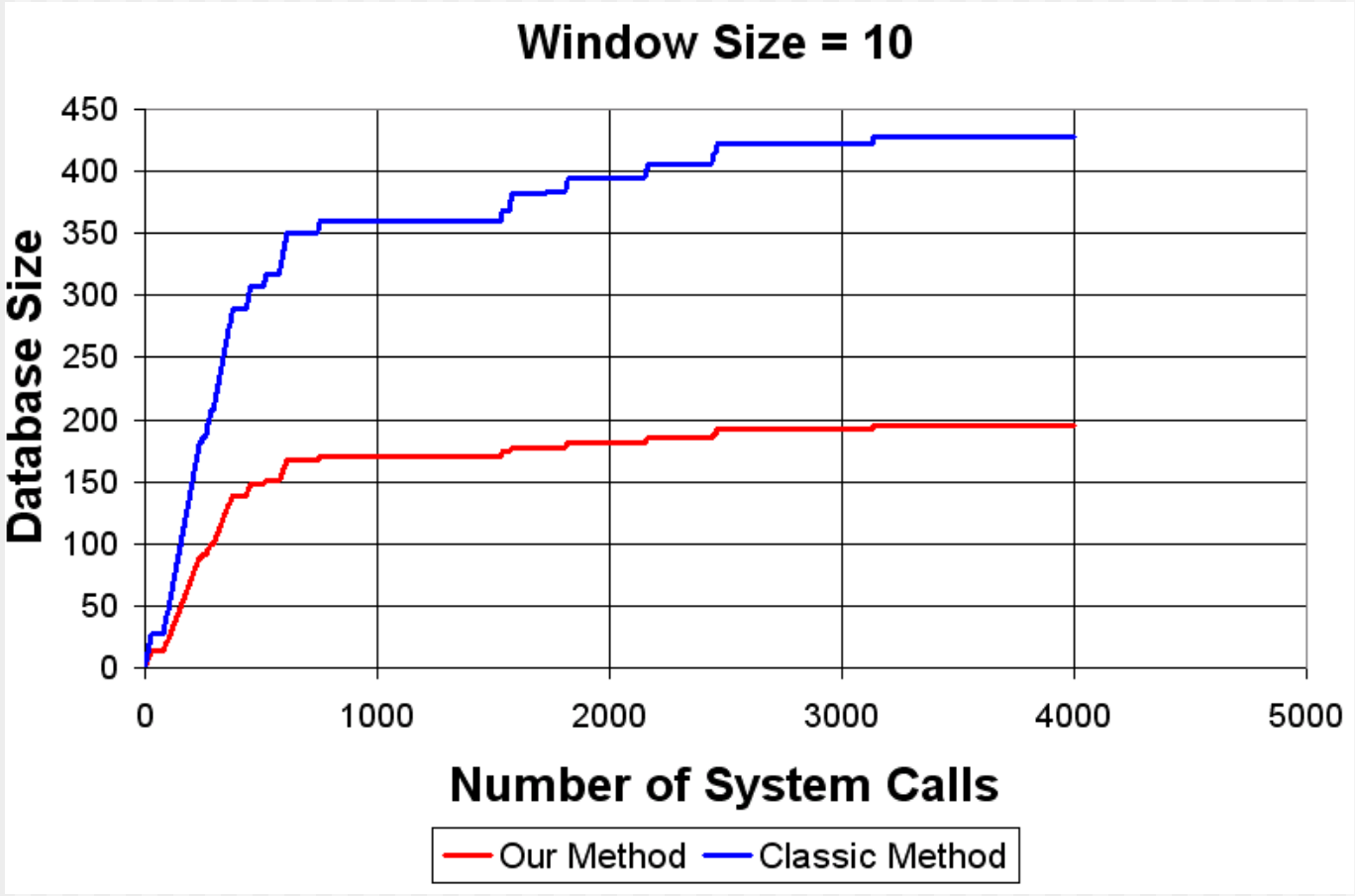
Applying SAA to sliding window

- Adding to database criterion is very strict – absolute equality between sequences.
- We can reduce the $NormDb$ size by replacing this criterion with more fuzzy.
- It can be similarity degree between sequences got after applying SAA.
- It would be great if the detection ability will not change.

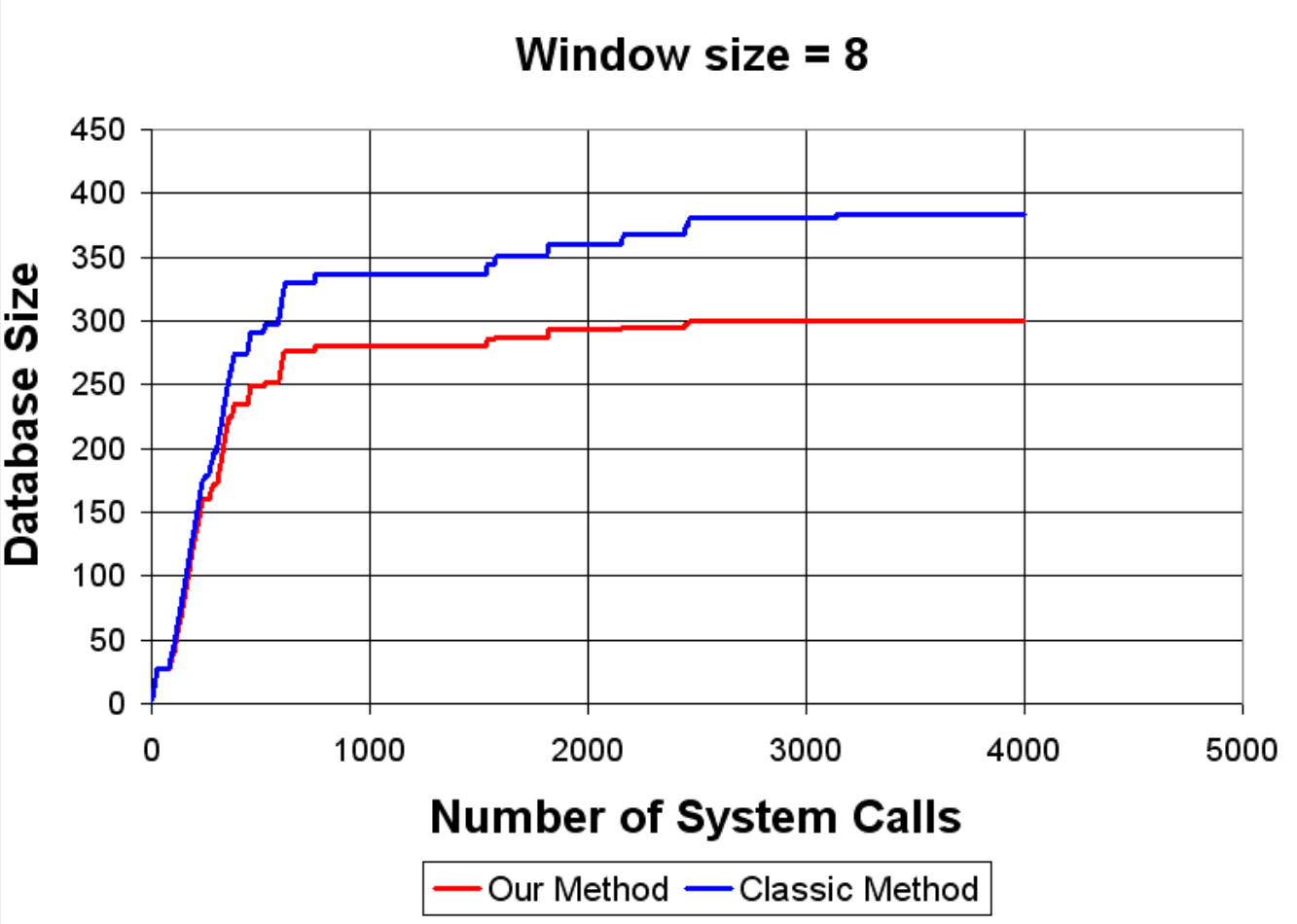
Sliding window method



Sliding window method



Sliding window method



Conclusion

- SAA are effective instrument that can be used for attack detection.
- SAA can be used to detect some kinds of mimicry attacks.
- SAA helped to reduce database size of sliding window method without affecting its detecting ability.
- SAA can be used in both anomaly and signature-based methods.

Future works

- Applying it not only to mimicry attack detection, but to detect mutations of computer viruses
- Implementation to detect not only mimicry attacks with no-ops
- Improvement of method complexity.