Anomalous System Call Detection

presented by Silviu Craciunas

Darren Mutz [1], Fredrik Valeur[1], Giovanni Vigna[1]
Christopher Kruegel[2]

[1]Department of Computer Science
University of California, Santa Barbara

[2]Department of Computer Science
Technical University of Vienna

22 May 2007

Outline

What is an IDS?

- The defender's problem:
  - The defender needs to plan for everything... the attacker needs just to hit one weak point
  - Being overconfident is fatal: King Darius vs. Alexander Magnus, at Gaugamela (331 b.C.)
- An IDS is a system, not a software!
- An IDS works on an information system, not on a network!

Two types of IDS

- Misuse-based
- Anomaly-based

Why system calls?

Sequences of system calls executed by running processes are a good discriminator of normal behavior.

Two types of IDS

- Misuse-based
- Anomaly-based

Why system calls?

Sequences of system calls executed by running processes are a good discriminator of normal behavior.

Two types of IDS

- Misuse-based
- Anomaly-based

Why system calls?

Sequences of system calls executed by running processes are a good discriminator of normal behavior.

Basics

- Application-specific analysis of individual system-calls
- Input consists of an ordered stream $S = \{s_1, s_2, ...\}$ of system call invocations recorded by the operating system
- Every $s \in S$ has $r_s, \langle a_1^s, a_1^s, ..., a_n^s \rangle$
- For every s a distinct profile is created

Learning

- The model is trained and the notion of normality is developed by inspecting samples
- Learning on-the-fly or learning from a training set

Important

Training phase must be as exhaustive and free from anomalous events as possible.

Detection

The task of a model is to return the probability of occurrence of an argument value based on the model's prior training phase. This value reflects the likelihood that a certain feature value is observed, given the established profile.

String length

- Arguments represent canonical filenames(open, stat, execv)
- Attacker must create a filename that triggers a format string vulnerability
- In such attacks the argument is a string of several hundred bytes

String length

Learning and detection

- Approximate the actual distribution of the lengths of a string argument
- Mean $\dot{\mu}$ and variance $\dot{\sigma}^2$ are approximated using μ and σ^2 for the lengths $l_1, l_2, \ldots, l_n$
- Probability for l: Cebyshev inequality $p(|x - \mu| > t) < \frac{\sigma^2}{t^2}$
- $p(l : l > \mu) = p(|x - \mu| > |l - \mu|) = \frac{\sigma^2}{(l-\mu)^2}$ *for* $l > \mu$

String character distribution

- Strings have a regular structure therefore we measure the frequencie values (not distribution)
- For a safe string the relative frequencies decrease in value, in malicious string the frequencies drop fast
- Idealized character distribution :

$$\mathfrak{ICD} : \mathfrak{D} \rightarrowtail \mathfrak{B} \ \textit{with} \ \mathfrak{D} = \{n \in \mathrm{N} | 1 \leq n \leq 256\},$$

$$\mathfrak{B} = \{p \in \mathfrak{R} | 0 \leq p \leq 1\},$$

$$\sum_{i=1}^{256} \mathfrak{ICD}(i) = 1.0$$

String character distribution

Learning and detection

- Learning phase :
  - Character distribution is stored for each argument string
  - ICD is calculated as an approximation of the average of all stored character distributions
- Detection :
  - Calculate the probability that the character distribution of an argument is a sample of the ICD

Structural inference

- Analyze the argument's structure, in our case it is the regular grammar that describes all legitimate values
- Conclude from this grammar by analyzing a number of legitimate strings

Example

Consider a simple open system call when an attacker exploits it trough a vulnerability and opens "/etc/passwd".

Structural inference

Learning and Detection

- Two choices: grammar that contains exactly the training data and a grammar that allows production of arbitrary strings
- First is too simple, second is too general
- Solution: generalize the grammar as long as it seems reasonable using probabilistic grammar
- The goal is to find a NFA(non-deterministic finite automata) of the probabilistic grammar that has the highest likelihood for the given data

Token finder

- Determines if the values of a argument are drawn from a limited set of possible alternatives
- The number of different argument values are bound
- Random values from type's value domain
- Decision between an enumeration and random identifiers can be made using the non-parametric Konglomorov-Smirnov variant
- Model returns 0 or 1 if he value is drawn from an enumeration depending on the correctness or in the case of random identifiers always 1

Classification

- A model m_i assigns an anomaly score as_i
- $C(as_1, as_2, \ldots, as_k, I) = \{normal, anomalous\}$
- In other systems C is a sum function, here, a **Bayesian network**

Definition

- A probabilistic graphical model that represents a set of variables and their probabilistic dependencies
- Formally, Bayesian networks are directed acyclic graphs whose nodes represent variables, and whose arcs encode the conditional dependencies between the variables
- $\forall$ vertexes v, $\nexists$ nonempty directed path that starts and ends in v

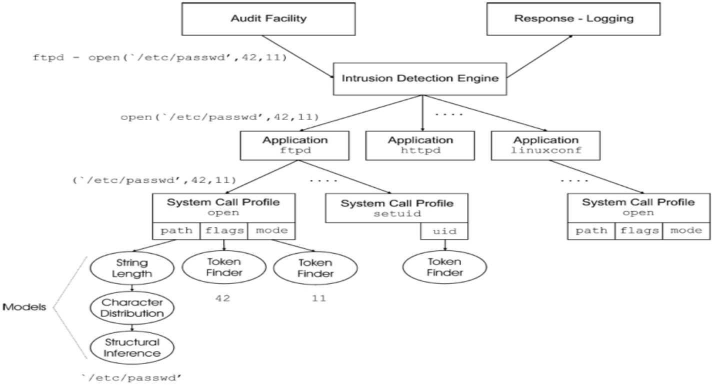
The mechanism

- Root node is a variable with two states: normal and anomalous
- One child node is introduced for each model (there might also be dependencies between models represented by connections)
- Additionally we have a confidence value represented by a node connected to the model node

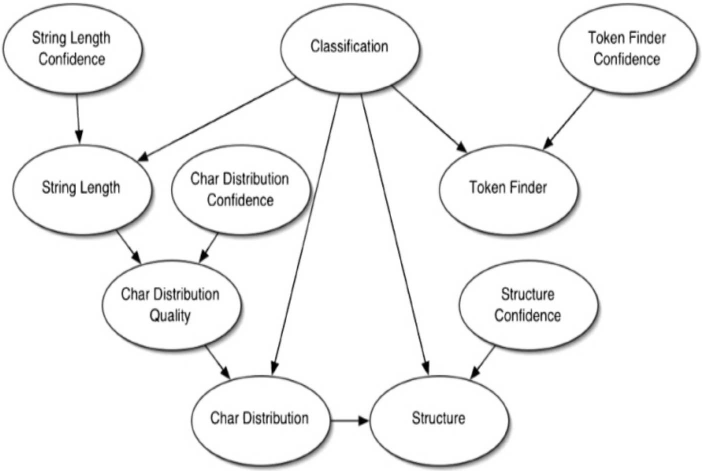
Overview

- Input from audit facilities(eg. Linux) or audit logs(eg. Solaris' BSM)
- Monitors security-critical applications(eg. setuid)
- For each program the IDS maintains data structure that characterizes the normal profile
- A profile consists of :
  - set of models for each argument
  - functions that calculates the anomaly scores

System architecture



Bayesian network for open and execve

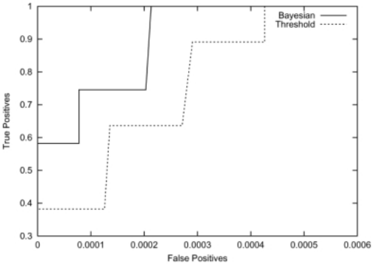


Classification Effectiveness

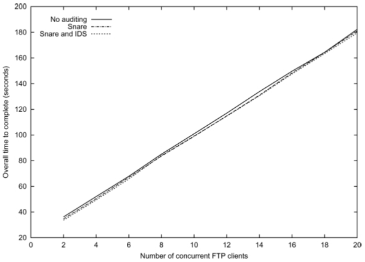
Application	Total System Calls	Attacks	Identified Attacks	False Alarms
eject	138	3	3 (14)	0
fdformat	139	6	6 (14)	0
ffbconfig	21	2	2 (2)	0
ps	4,949	14	14 (55)	0
ftpd	3,229	0	0	14
sendmail	71,743	0	0	8
telnetd	47,416	0	0	17
Total	127,635	25	0	39

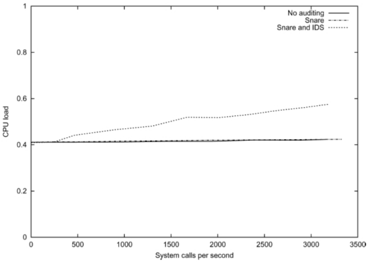
Classification Effectiveness

Application	Sequences FN	Sequences FP	Syscall Bags FN	Syscall Bags FP	K-Nearest FN	K-Nearest FP	Cluster FN	Cluster FP	Our System FN	Our System FP
eject	1	1	1	1	2	1	0	1	0	0
fdformat	2	0	2	0	0	0	0	0	0	0
ffbconfig	0	0	0	0	0	0	0	0	0	0
ps	0	12	0	0	0	47	12	25	0	0
ftpd	0	21	0	15	0	21	0	20	0	14
sendmail	0	75	0	1	0	89	0	106	0	8
telnetd	0	99	0	99	0	21	0	6	0	17
Total	3	208	3	116	2	179	12	158	0	39



System Efficiency





Conclusion

- Learning based algorithm
- Includes system call arguments
- Combining multiple anomaly scores using Bayesian networks
- Outperforms the top 4 learning based IDS on a well known intrusion detection evaluation data set
- Low computational and memory overhead

Conclusion

- Learning based algorithm
- Includes system call arguments
- Combining multiple anomaly scores using Bayesian networks
- Outperforms the top 4 learning based IDS on a well known intrusion detection evaluation data set
- Low computational and memory overhead

Conclusion

- Learning based algorithm
- Includes system call arguments
- Combining multiple anomaly scores using Bayesian networks
- Outperforms the top 4 learning based IDS on a well known intrusion detection evaluation data set
- Low computational and memory overhead

Conclusion

- Learning based algorithm
- Includes system call arguments
- Combining multiple anomaly scores using Bayesian networks
- Outperforms the top 4 learning based IDS on a well known intrusion detection evaluation data set
- Low computational and memory overhead

Conclusion

- Learning based algorithm
- Includes system call arguments
- Combining multiple anomaly scores using Bayesian networks
- Outperforms the top 4 learning based IDS on a well known intrusion detection evaluation data set
- Low computational and memory overhead

Thank you

Thank you!
Any questions? *(Hopefully NOT!)*

Bayesian network validation

- If there is a causal relationship between models $\Rightarrow \exists$ corelation between model scores.
- Calculate correlation value for all pairs of models