Detecting Stealthy Backdoors with Association Rule Mining

Stefan Hommes, Radu State, Thomas Engel

Detecting Stealthy Backdoors with Association Rule Mining

Stefan Hommes, Radu State, Thomas Engel

University of Luxembourg, SnT
6, rue R. Coudenhove-Kalergi, L-1359 Luxembourg
{stefan.hommes,radu.state,thomas.engel}@uni.lu

Abstract. In this paper we describe a practical approach for detecting a class of backdoor communication channel that relies on port knocking in order to activate a backdoor on a remote compromised system. Detecting such activation sequences is extremely challenging because of varying port sequences and easily modifiable port values. Simple signature-based approaches are not appropriate, whilst more advanced statistics-based testing will not work because of missing and incomplete data. We leverage techniques derived from the data mining community designed to detect sequences of rare events. Simply stated, a sequence of rare events is the joint occurrence of several events, each of which is rare. We show that searching for port knocking sequences can be reduced to a problem of finding rare associations. We have implemented a prototype and show some experimental results on its performance and underlying functioning.

Keywords: backdoor, association rule mining, cd00r

1 Introduction

The detection of stealthy backdoor communications is challenging because of a lack of proven methodologies and targeted tools. A typical backdoor makes it possible for an attacker to connect at will and perform actions on compromised machines. While such backdoors operate through a shell command line interface operated over TCP ports opened on a compromised machine, stealthy backdoors will avoid detection by enabling temporary shells for short periods. This is done by promiscuously intercepting network traffic and activating a remote shell when a given sequence of predefined IP packet patterns occurs. The most famous backdoor that implements this technique is cd00r [2], where a series of successive IP packets is used to trigger a remote shell activation. The underlying principle is quite simple. Several IP packets are sent to the compromised machine. The port is activated by a defined sequence of TCP SYN packets, followed by the chosen port number that is then used to access the compromised host. If this sequence of predefined destination ports occurs, then a basic shell is opened on the machine. Consequently, this shell runs over a port only during the time interval the attacker is interacting with the machine. Until now, detecting such

a backdoor was impossible because the specific series of destination ports can easily be changed and extended by the attacker. In this paper we propose a method for detecting such backdoors. We leverage a data mining technique for searching for associations between rare events and demonstrate its performance and efficiency on several datasets and attack scenarios. The major contributions of our paper are:

- We describe a practical approach for detecting stealthy backdoors that rely on port knocking mechanisms.
- We propose a methodology to set the parameters (minimum support, confidence and interest) used to search for rare associations.
- We have implemented and assessed our prototype with respect to existing known backdoors.

Our paper is structured as follows: we start in section 2 with an overview of related research work. We give an introduction to stealthy backdoors in section 3 in order to make the paper self-contained. Section 4 describes the method used to search for rare events. We present a first set of experimental results in section 5 in order to validate our approach. Finally, section 6 summarizes the paper and suggests future work.

2 Related Work

Agrawal [1] introduces association rule mining to extract interesting correlations and patterns from databases. The most popular algorithms such as Apriori are able to identify frequent itemsets, in which the relationship between items can be expressed in the form of association rules. A much less explored area is infrequent itemset mining, which deals with the problem of finding rare itemsets in a huge database. Rare association rules must be separated from items that occur together simply by coincidence, which is difficult, since a low support threshold leads to a huge number of possible rules. A variable support threshold is proposed by [3], while imposing structure constraints is proposed by Apriori-Inverse [7].
Although no academic work has specifically addressed the automated detection of port knocking sequences, relevant prior work in detecting network anomalies does exist. The most similar to our application is the detection of port scanning, albeit that there is a significant difference between port knocking and scanning. At a first glance, both are similar, since in port knocking, the initial TCP packets might look like a scanning attempt. However, scanning detection relies on a much larger set of unsuccessful TCP/IP establishment requests in order to infer scanning activity. The Threshold Random Walk (TRW) portscan detection [6] requires more than five unsuccessful connections in order to detect a scan. Secondly, an attacker can easily spoof the source IP addresses in the port knocking packets and thus evade any approach oriented at identifying a particular source IP address. Similarly, [15] relies on additional traffic volumes to detect scanning activities, but again due to spoofing, its suitability for our purposes is very

limited. While anomaly detection methods relying on protocol specific statistics [10], [16], and [11] are useful when the quantity of data is reasonable, detecting anomalies resulting from only five or fewer packets is impossible.

3 Concept of Backdoor cd00r

Cd00r demonstrates a hidden Linux backdoor that cannot be detected by network scanners and intrusion detection systems (IDS). It is not visible in the network socket table until the backdoor is activated by a sequence of TCP, UDP or ICMP packets. Afterwards, a shell or other tools (e.g. netcat[1]) can be used to access the compromised machine. This is possible due to the fact that cd00r uses the libpcap library to listen on raw sockets. Therefore, applications like netstat or nmap that listen on higher-level sockets cannot detect that the port is being monitored [4].
In addition, many low-end firewalls lack advanced deep packet inspection features and thus do not block modified and illegally formed packets used in attempts to connect to the compromised machine [5]. For a network administrator it is therefore very difficult to detect or prevent the use of such backdoors. On the other hand, the backdoor could also provide a secure way to defend a system against attacks, since critical services can be protected by the fact that the port is not visible to the outside.
In our scenario, the activation of the backdoor uses a sequence of TCP SYN packets that must be sent to a list of ports in the correct order (knocking). Furthermore, a hard-coded port is not needed for the connection to the compromised machine since we can choose it dynamically by sending the chosen port after the activation sequence. The following example demonstrates a complete session to activate and communicate with the backdoor using telnet:

```
t="telnet host.victim.com"
$t 999; $t 888; $t 777; $t 666; $t 555; $t 1234; sleep 1; $t 1234
```

The ports 999, 888, 777, 666 and 555 are used to activate, and the port 1234 to define where the shell should be spawned. After waiting for a short time (sleep 1) for the shell to be started on the target machine, we can connect with telnet and have access to the system. The implementation of the backdoor was done using the code of [2]. The original cd00r implementation is using a hard coded listening port and can be found at [14]. The implementation of a similar backdoor called Sadoor is available at [13]. Advanced methods of port knocking with secret sharing that need a group of people to open ports is proposed in [12].

4 Rare Association Rule Mining

Finding association rules as a common method in data mining was introduced by Agrawal, Imielinski, and Swami in 1993 [1]. Analysing supermarket transactions

[1] http://netcat.sourceforge.net/

was one of the original motivations for finding association rules, since this can reveal information about products that customers usually buy together. Just as information about frequent itemsets (e.g. bread, butter and jam) is interesting in some applications, finding *infrequent* itemsets is useful in finding events that rarely occur and can be the result of a failure or attack. A valid rule can be defined as the combination of k-itemsets that appear only seldom in a dataset transaction, but not simply by coincidence, and which would have been normally pruned out when using classical association mining algorithms such as Apriori [1].
The following sections describe the algorithm that discovers association rules among infrequent items and is used to find the knocking sequence of TCP ports for the backdoor described in the previous section.

4.1 Terminology

A TCP port is considered to be a random variable I with event set $\mathbb{E}(I) = \{0, ..., 65535\}$. All recorded port accesses are saved in a database D and are divided into windows w of a defined size w_s. For each window we get $w = \{i_1, i_2, ..., i_{w_s}\}$. Let X, Y be itemsets of ports that occur together in a window, the resulting association rule is further $X \Rightarrow Y$ with $X, Y \subset \mathbb{E}(I)$ and $X \cap Y = \varnothing$. The percentage of windows in the dataset that contain the rule can be described by the *support*,

$$\text{supp}(X \cup Y) = P(X \cup Y) \tag{1}$$

The ratio of windows that contain X and also contain Y is defined as the *confidence*:

$$\text{conf}(X \Rightarrow Y) = P(Y|X) = \frac{P(X \cup Y)}{P(X)} \tag{2}$$

In order to find interesting association rules, the minimum threshold for support (*minSupp*) and confidence (*minConf*) must be defined by the analyst in advance and depends of the dataset. A common problem when finding an infrequent itemset is that, since both sets rarely occur, the minimum support threshold (*minSupp*) has to be set very low. This results in a huge number of possible combinations of items and is known as the rare item problem [9]. In general, rare association rule mining requires low minSupp but high minConf to reduce the overall number of rules as a first step. Since the resulting number of rules can still be very high, uninteresting rules must be filtered out, and we rely on the pruning strategy described by Wu [17].
Since X and Y are independent if $P(X \cup Y) = P(X) \cdot P(Y)$, Wu defined an interest function (*minInte*) that is shown in equation 3. By comparing the interest value with a minimum interest threshold set by the analyst, this interest function helps to reduce the number of rules and removes itemsets which are independent of each other and are not considered as valid rules. The correlation function in

equation 4 determines the dependence of two itemsets X, Y with three possible cases.

$$\text{Interest}(X,Y) = |\text{supp}(X \cup Y) - \text{supp}(X)\text{supp}(Y)| \tag{3}$$

$$\text{Correlation}(X,Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)\text{supp}(Y)} = \begin{cases} 1 & \text{independent} \\ > 1 & \text{pos. correlated} \\ < 1 & \text{neg. correlated} \end{cases} \tag{4}$$

A positive correlation means that the occurrence of one itemset increases the occurrence of the other itemset, while a negative correlation means that the occurrence of one itemset discourages the occurrence of the other. To determine if a found rule is above the minimum confidence threshold (minConf), Wu defined $\text{conf}(X \rightarrow Y)$ to be $\text{CPIR}(Y|X)$:

$$\text{CPIR}(Y|X) = \frac{\text{supp}(X \cup Y) - \text{supp}(X)\text{supp}(Y)}{\text{supp}(X)(1 - \text{supp}(Y))} \tag{5}$$

4.2 Algorithm

The pseudocode of the matrix-based scheme is displayed in algorithm 1. We have relied on the description given in [8]. As a first step, we scan the whole dataset and store all port numbers that have a support count bigger than minSupp in a matrix, Inf1 (see lines 1 to 4). In lines 5 to 15, we check whether each port in every window is infrequent by checking the Inf1 matrix, and map if it is included to a new matrix, temp. If a window in temp contains more than one element, we calculate all possible combinations and increase the support count of each combination in a new matrix called Inf2.
In lines 16 to 24, we calculate the interest value (equation 3) for every itemset in Inf2 that has a support count bigger than one. If the interest is below minInte, we remove the itemset from the matrix. For the remaining itemsets, we determine (lines 25 to 32) for each if the correlation (equation 4) is greater or equal than one and if CPIR (equation 5) is larger than minConf. This last step is needed to determine the correct order of the elements in the itemset.

5 Experimental Results

The experiments were run on a dual-core 2.80 GHz Intel PC and 4 GB of RAM with an Ubuntu 64 bit OS. The backdoor was installed on an Ubuntu VM with the network card running in bridged mode. We evaluated our results on two datasets which contain the complete network traffic between the attacker and the target host. There were five complete connections to the backdoor during the period covered by each dataset. Each connection consisted of a port-knocking sequence, followed by a user-defined port for the shell. After opening the port and

Algorithm 1 Matrix-based scheme (Zhou, Yau [8])

Require: database D, minSupp, minConf, minInte, association rules AR = ∅
scan the database D and find all infrequent 1-itemsets (Inf1)
Item ← {a matrix used to store information of all items in D}
Item.index ← the index value of infrequent item in Inf1, frequent items with a index value of -1;
Infk← {matrices used to store support counts of infrequent k-itemsets, where $k > 1$};
scan database D a second time
for each window $w_i \in$ D **do**
  for each item i ∈ window w_i **do**
    if i.index $\neq -1$// identify infrequent items **then**
      map i.index into Temp
    end if
  end for
  if the number of items in Temp is greater than 1, **then**
    find all combinations of these values and increase support count of each combination
  end if
end for
for each k-itemset I ∈ Infk **do**
  if I.count≥ 1 **then**
    for ∀ itemsets $X, Y, X \cup Y = I$ and $X \cap Y = \varnothing$ **do**
      if interest(X,Y)< minInte **then**
        Infk ← Infk-{I};
      end if
    end for
  end if
end for
for each infrequent k-itemsets of interest $X \cup Y \in Infk$ **do**
  if correlation$(X, Y) > 1$ && CPIR$(Y|X) \geq$ minConf **then**
    $AR \leftarrow \{X \Rightarrow Y\}$
  end if
  if correlation$(X, Y) > 1$ && CPIR$(X|Y) \geq$ minConf **then**
    $AR \leftarrow \{Y \Rightarrow X\}$
  end if
end for
return AR

connecting to the backdoor, a text document was saved on the target machine and the connection was closed.
To determine the optimal parameters for the rule mining algorithm, we used an analytical approach based on statistical parameters that are calculated from the dataset beforehand and which are displayed in table 1.

Table 1. Dataset statistics

	Dataset 1	Dataset 2
Total number of TCP packets/ports t	19609	50787
Involved port numbers u	559	804

5.1 Analytical determination of system parameters

In order to set the correct values for minSupp, minConf and minInte, we have to determine the number of involved port numbers u and the number of total ports t in the dataset. The resulting number of windows is defined by $l = t/w_s$, where the window size w_s has to be chosen by experiment. We had promising results with $w_s = 1000$, which gives a high probability that the port sequence is completly included in a single window. If the number of backdoor activations is known, the optimal minSupp can be determined by

$$supp = \frac{\text{number of knocking sequences}}{\text{number of windows } l} \tag{6}$$

To determine the optimal minSupp when the number of backdoor activations is unknown, we propose the following analytical approach. The probability that a certain port is included in one window is given by equation 7.

$$p = P(\text{port} \in \text{window}) = 1 - \left(1 - \frac{1}{u}\right)^{w_s} \tag{7}$$

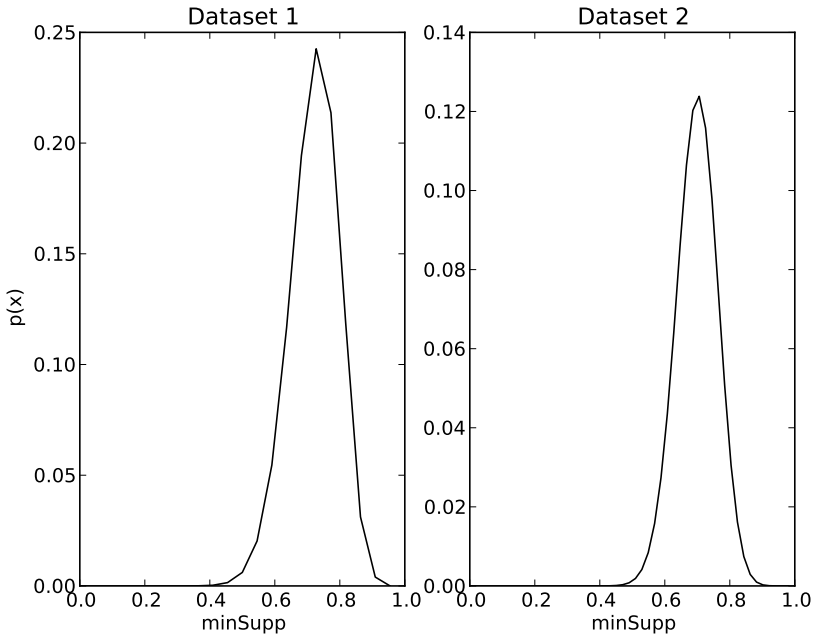
The probability that a port is in exactly x windows can be calculated as a binominal distribution and is in equation 8. In order to simplify the calculation of the permuatation, logarithmic values were used in the implementation of the algorithm.

$$p(x) = \binom{l}{x} \cdot \left(1 - \left(1 - \frac{1}{u}\right)^{w_s}\right)^{x} \cdot \left(\left(1 - \frac{1}{u}\right)^{w_s}\right)^{l-x} \tag{8}$$

5.2 Results

In order to evaluate the results on both datasets, we need to set the correct parameters for minSupp, minInte and minConf. In the first case, we see from

Fig. 1. Probability that a port is in exactly x windows



equation 6 that, for datasets 1 and 2, the optimal minSupp values are approximately 0.3 and 0.1 respectively for a window size of $w_s = 1000$. Since the number of activations is unknown in a realistic scenario, we can choose minSupp by calculating the probabilities of the binominal distribution for different support thresholds from equation 8, giving the results displayed in figure 1. The chosen minSupp should be in the range that lies between 0 and the gradient of the probability function.

Determining the correct minInte was done in an experiment after choosing minSupp of 0.3 for dataset 1 and 0.1 for dataset 2. Table 2 displays the found rules in the dataset after increasing minInte from zero to one, which results in a decrease of rules. To reduce the number of uninteresting rules and to assure that two itemsets are not independent, the authors propose a minInte that is approximatly at 0.17 for dataset 1 and 0.07 for dataset 2.
Based on the different settings of the minSupp threshold, the results for both datasets are displayed in table 3 and table 4. We used *Precision* and *Recall*, two statistical measures that determine the quality of a binary classification test. In our scenario, the knocking sequence of the backdoor results in 20 possible rules (e.g [555, 666], [777, 888]). The recall is defined as the number of correctly found rules divided by the number of all possible rules. The precision defines the quality of the classification and can be calculated from the number of correctly

found rules divided by the number of found rules. The results achieve a recall rate of 70% combined with a high precision of approximatly 80%.

Table 2. Detected rules for different interest values (minInte)

minInte	Rules in Dataset 1	Rules in Dataset 2
0.00	10437	12847
0.01	10437	12847
0.02	10437	1639
0.03	10437	1639
0.04	10437	287
0.05	10437	287
0.06	1586	14
0.07	1586	14
0.08	1586	0
0.09	1586	0
0.10	1586	0
0.15	205	0
0.20	16	0
0.25	14	0
0.30	0	0
0.35	0	0

Table 3. Results for dataset 1

Parameters	Recall	Precision
minSupp = 0.30	(14/20) = 0.70	(14/16) = 0.88
minSupp = 0.50	(14/20) = 0.70	(14/16) = 0.88
minSupp = 0.70	(14/20) = 0.70	(14/17) = 0.82
minInte = 0.17		
minConf = 0.90		

6 Conclusion and Future Work

In this paper we have shown a practical and efficient approach to detect port-knocking sequences in network traffic. To the extent of our knowledge, very few previous papers have tackled this issue. This might be due to the inherent complexity of the task as well as to a limited deployment of backdoors based on port knocking. As a matter of fact, very little is known about the operational management of such backdoors. Since no existing tool detects them, such backdoors might have been in place for a long time without being detected by existing monitoring solutions. We have designed a monitoring tool that relies

Table 4. Results for dataset 2

Parameters	Recall	Precision
minSupp = 0.10	(14/20) = 0.70	(14/14) = 1.00
minSupp = 0.40	(14/20) = 0.70	(14/22) = 0.63
minSupp = 0.70	(14/20) = 0.70	(14/32) = 0.44
minInte = 0.07		
minConf = 0.90		

on data mining methods targeted at identifying sequences of rare events and we have shown in this paper that the proposed approach is viable. We have illustrated the performance of our tool on two datasets generated in a controlled environment, where we could precisely assess the true positive and false negative rates. We are aware that these datasets might not be representative for a typical network. As a matter of fact, we also used larger datasets from large campus networks, but the major issue was that the obtained results could not be easily validated. This would have required us to have forensic access to the suspicious machine, which was impossible from an administrative point of view. The major challenge in finding rare association rules in network traffic is due to the fact that the required low minimum support leads to a huge number of rules. We have proposed a set of simple rules that rely on pruning and a parameter-setting method. An analytical approach is needed to achieve a good understanding of the probabilities that a certain itemset will occur, and how to set the parameters of the algorithm. If this requirement is fulfilled, the proposed concept is able to find the port sequence of stealthy backdoors. Datasets from a longer time period should achieve even better results, because other infrequently accessed ports that can be considered as noise will occur more often so no longer be infrequent. The developed source code is available on request. We look forward to extend our approach to a broader set of network data that ranges from firewall logs to passive DNS monitoring.

Acknowledgement

The present project is supported by the National Research Fund, Luxembourg.

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. pp. 207–216. ACM
2. FunOverIP: cd00r knocking backdoor (improved) (2011), `http://funoverip.net/2011/03/cd00r-knocking-backdoor-improved/`
3. Hahsler, M.: A model-based frequency constraint for mining associations from transaction data. Data Min. Knowl. Discov. 13, 137–166 (September 2006)
4. Hay, G.: Extending the packet coded backdoor server to netcat relays on relatively high-bandwidth home networks. Tech. rep., SANS (2001)

5. Jonathan, Y.: Use port knocking to bypass firewall rules and keep security intact (2005), http://www.techrepublic.com/article/use-port-knocking-to-bypass-firewall-rules-and-keep-security-intact/5798871
6. Jung, J., Paxson, V., Berger, A.W., Balakrishnan, H.: Fast portscan detection using sequential hypothesis testing. In: In Proceedings of the IEEE Symposium on Security and Privacy (2004)
7. Koh, Y.S., Rountree, N.: Finding sporadic rules using apriori-inverse. Lecture Notes in Computer Science, vol. 3518, pp. 97–106. Springer (2005)
8. Koh, Y.S., Rountree, N.: Rare Association Rule Mining and Knowledge Discovery: Technologies for Infrequent and Critical Event Detection. Information Science Reference - Imprint of: IGI Publishing, Hershey, PA (2009)
9. Liu, B., Hsu, W., Ma, Y.: Mining association rules with multiple minimum supports. In: Knowledge Discovery and Data Mining. pp. 337–341 (1999)
10. Mahoney, M., , Mahoney, M.V., Chan, P.K.: Learning rules for anomaly detection of hostile network traffic. In: Proc. of International Conference on Data Mining (ICDM. pp. 601–604 (2003)
11. Marchetti, M., Colajanni, M., Manganiello, F.: Identification of correlated network intrusion alerts. In: Proc. of the 3rd IEEE International Workshop on Cyberspace Safety and Security (CSS 2011)
12. Miklosovic, S.: Pa018 - term project - port knocking enhancements (2011), http://www.portknocking.org/view/resources
13. Nyberg, C.M.: Sadoor. http://packetstormsecurity.org/UNIX/penetration/rootkits/index7.html
14. Phenoelit: cd00r.c - packet coded backdoor (2000), http://www.phenoelit-us.org/stuff/cd00r.c
15. Schechter, S.E., Jung, J., Berger, A.W.: Fast detection of scanning worm infections. In: Proceedings of the 7 th International Symposium on Recent Advances in Intrusion Detection (RAID. pp. 59–81 (2004)
16. Valdes, A., Skinner, K.: Adaptive, model-based monitoring for cyber attack detection. In: Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection. pp. 80–92. RAID '00, Springer-Verlag (2000)
17. Wu, X., Zhang, C., Zhang, S.: Efficient mining of both positive and negative association rules. ACM Trans. Inf. Syst. 22, 381–405 (July 2004)