Finding Corrupted Computers Using Imperfect Intrusion Prevention System Event Data

Danielle Chrun[1], Michel Cukier[1], and Gerry Sneeringer[2]

[1] Center for Risk and Reliability, University of Maryland
College Park, Maryland 20742-7531
[2] Office of Information Technology, University of Maryland
College Park, Maryland 20742-7531
{chrun,mcukier,sneeri}@umd.edu

Abstract. With the increase of attacks on the Internet, a primary concern for organizations is how to protect their network. The objectives of a security team are 1) to prevent external attackers from launching successful attacks against organization computers that could become compromised, 2) to ensure that organization computers are not vulnerable (e.g., fully patched) so that in either case the organization computers do not start launching attacks. The security team can monitor and block malicious activity by using devices such as intrusion prevention systems. However, in large organizations, such monitoring devices could record a high number of events. The contributions of this paper are 1) to introduce a method that ranks potentially corrupted computers based on imperfect intrusion prevention system event data, and 2) to evaluate the method based on empirical data collected at a large organization of about 40,000 computers. The evaluation is based on the judgment of a security expert of which computers were indeed corrupted. On the one hand, we studied how many computers classified as of high concern or of concern were indeed corrupted (i.e., true positives). On the other hand, we analyzed how many computers classified as of lower concern were in fact corrupted (i.e., false negatives).

Keywords: Security Metrics, Empirical Study, Intrusion Prevention Systems.

1 Introduction

With the increase of attacks on the Internet, a primary concern for organizations is how to protect their network. To do so, organizations monitor their traffic using security devices such as intrusion detection systems or intrusion prevention systems. The monitored activity provides some insight into an organization's security and identifies potentially corrupted computers. While in some organizations the quantity of monitored traffic is manageable, it becomes a hassle to analyze security data for large organizations. For example, intrusion prevention systems could record thousands of alerts per day and the security team cannot investigate every alert. Moreover, although intrusion prevention systems are aimed at detecting and blocking malicious activity, they also raise false alarms. Due to 1) the potentially large quantity of data to deal with, and 2) the number of

M.D. Harrison and M.-A. Sujan (Eds.): SAFECOMP 2008, LNCS 5219, pp. 221–234, 2008.

false alarms, it is of main interest to organize the generated alerts and to extract information from the collected data that would be useful to the security team.

This paper presents a method to retrieve useful information for the security team from data collected by an intrusion prevention system (IPS). The method consists in identifying potentially corrupted computers inside the organization and ranking them according to three metrics: the coefficient of consecutiveness indicating during how many consecutive weeks IPS alerts were observed, the number of weeks during which alerts were raised and the number of distinct attack types. Based on these metrics, potentially corrupted computers can be ranked. We will show that the proposed method helps the security team gaining some insight into the organization's security. The introduced method is evaluated for data collected at a large organization of about 40,000 computers. The evaluation is based on the judgment of a security expert of which computers were indeed corrupted. On the one hand, we studied how many computers classified as of high and medium concern were indeed corrupted (i.e., true positives). On the other hand, we analyzed how many computers classified as of low concern were in fact corrupted (i.e., false negatives).

The remainder of the paper is structured as follows. Section 2 describes the related work on data analysis of security logs. Section 3 introduces the concepts relative to IPSs. Section 4 defines the method. Section 5 presents the evaluation of the method. We provide conclusions in Section 6.

2 Related Work

A lot of research focuses on analyzing security logs for security assessment. To face the possibly high quantity of data to analyze, a common step is to reduce data before analyzing it. [1] describes an architecture to analyze distributed darknet traffic: first, collected data on attacks are filtered; secondly, forensics is used to analyze the reduced data. [2] focuses on analyzing data of a denial of service. In order to study the traffic volume per protocol, a categorization of the collected network traffic by protocol was made.

Analyzing large amounts of security data becomes an emerging task in the intrusion detection field. Indeed, intrusion detection systems face two main issues: 1) a high number of alarms can be raised and 2) there can be many false alarms among them. Thus, the objective is to decrease the number of false alarms. Research was conducted to retrieve normal behavior (i.e., traffic that is not malicious) from the dataset using several techniques: time series [3], data mining [4, 5, 6, 7, 8, 9, 10 and 11] and correlation [6, 12, 13, 14, 15 and 16]. A common practice is to use historical data to define normal behavior so that future alarms can be handled more efficiently. Data mining techniques can be used to achieve this goal. However, research projects differ in the data mining technique used: association rules [10], frequent episode rules [4, 9], classification [11] or clustering [5, 6, 7, 8 and 9]. A commonly used method in intrusion detection is alert correlation. [13] defines a model for intrusion detection alert correlation and presents three examples of correlation: aggregation of alerts referring to a single targeted host, aggregation of alerts referring to hosts vulnerable to an attack occurrence and aggregation according to alerts similarities (such as alerts caused by the same event or referring to the same vulnerabilities). [5 and 6] introduce a cooperative intrusion detection framework in which functions to manage, cluster, merge and correlate alerts were implemented. The objective was to correlate alerts to generate more global alerts and discard false alarms.

In [11], the authors present the Adaptive Learner for Alert Classification (ALAC) system. ALAC is a system to reduce false positives in intrusion detection systems and relies on two elements: 1) expert judgment and 2) machine learning techniques. An analyst classifies alerts as true positives or false positives. Then, ALAC autonomously processes alerts that have been classified by the analyst. The accuracy of ALAC is as good as the quality of the analyst's classification.

3 On the Use of Intrusion Prevention System Event Data

3.1 Approach

Many organizations use security devices to monitor their network activity. The quantity of data collected per day can be so substantial that every event identified by a security device cannot be investigated by the security team. Hence, retrieving meaningful information from the collected data on the malicious activity would give a more detailed insight to security administrators into the network's security. The main issue is that the data currently collected are far from being perfect. For example, the data collected by security devices, such as intrusion prevention systems (IPSs), might contain alerts for activity that is not malicious (i.e., false positives) and might not detect some malicious activity (i.e., false negatives). Moreover, they will not include new attacks in the case of signature-based IPSs. They often rely on the trust we have in the security devices and the vendors. No ground truth is provided. Details are lacking on the meaning of the data and how they are produced (the security devices are black boxes for which vendors only release few details).

Two approaches are then possible. The first one is to work on obtaining datasets clean enough so that accurate security estimations are possible. The second one is to accept that the dataset is imperfect but that useful information regarding an organization's security can be retrieved. In this paper, we adopt the second approach.

In this paper, we provide a method to extract useful information from IPS event data. The suggested method aims at extracting a list of potentially corrupted organization computers that would then be handled by the security team. Those computers manifest in the IPS dataset as the potential source of attacks. The dataset might not only contain attackers who willingly launch attacks. It might also include computers that may not have been fully patched. Once the list of suspected computers is identified, the security team can make a decision regarding these computers. For example, a decision could consist in blocking the IP address from the network until the computer is cleaned.

3.2 Intrusion Prevention Systems

An IPS is a security device that monitors malicious activity and reacts in real-time by blocking a potential attack. An IPS is considered as an extension of an intrusion detection system (IDS). An IDS is a passive device that monitors activity whereas an IPS is an active device that blocks potential malicious activity. For our study, we focus on signature-based IPSs: the IPS blocking decision relies on a set of signatures that are regularly released by the vendor as attacks are newly discovered on the Internet. When characteristics of an attack match the ones of a defined signature, the attack is blocked and an alert is recorded in the IPS logs.

We assume that the IPS is located at the edge of the organization. In other words, the IPS monitors 1) malicious activity originating inside the organization and targeting outside computers, 2) malicious activity originating outside the organization and targeting organization computers.

We define an alert in the IPS dataset as a source IP address (SIP/attacker) attacking a destination IP address (DIP/target) with a certain type of attack (signature) at a given time.

3.3 Dataset: Assumptions

As previously mentioned, the IPS dataset has several issues. We have not evaluated the IPS and thus do not know how many false positives and false negatives the IPS produces. Moreover, since the IPS is a signature-based device, new attacks will not be detected nor blocked.

Furthermore, the dataset does not include the case where a computer inside the organization attacks another computer inside the organization. The IPS is located at the edge of the organization so it cannot detect traffic within the organization. Besides, this study solely focuses on computers with static IP addresses.

Finally, we cannot prove that a blocked attack would have been harmful to the targeted computer. Indeed, for an attack to be successful, the targeted computer should have the associated vulnerability. We have scanned several computers for which an IPS alert was raised and noticed that in many cases the vulnerability associated with the alert was not present. This means that even without the IPS, the attack would not have been successful. This also indicates that the IPS identifies and detects an attack in its early stage preferring to block attacks that would not have been successful instead of not blocking a potentially successful attack.

4 Method

The next sub-sections present the method to identify potentially corrupted organization computers. First, we define three metrics to characterize the activity in the IPS dataset. Then, we present the method for ranking the potentially corrupted computers according to the three metrics values.

4.1 Metrics

A computer is of main concern to the security team if 1) it appears often in the IPS dataset as the source of an attack, and 2) it launches a wide range of different attack types. Therefore, we introduce the following metrics for a computer: 1) a coefficient of consecutiveness of the number of weeks for which at least one alert was raised, 2) the number of weeks for which at least one alert was raised, and 3) the number of different signatures (i.e. attack types) associated to the computer. We defined these metrics that we believe are appropriate for attackers. These metrics might be less relevant for targets (computers under attack).

4.1.1 Coefficient of Consecutiveness

Computers that appear in the IPS dataset for many consecutive weeks are of main concern for the organization's security team, seeming to indicate that a computer is

launching attacks during several consecutive weeks and has not been checked. We define the coefficient of consecutiveness as:

$$Cons = Week/(Max - Min + 1)$$

where *Max* is the identifier of the last week when the computer appears in the dataset, *Min* is the identifier of the first week, and *Week* is the number of distinct weeks. The consecutiveness factor is positive and the maximum value is 1. Let us consider a computer that appears in the IPS dataset at weeks 2, 3, 6, 8, 9, among 10 weeks of observation (Figure 1). In this case, *Max* = 9, *Min* = 2 and *Week* = 5. The consecutiveness factor is: 5/(9-2+1) = 0.625.

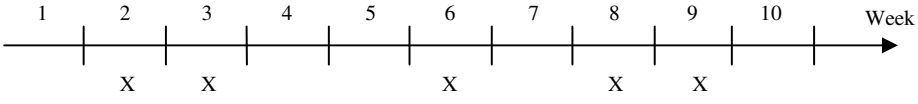

Fig. 1. Consecutiveness Factor

The closer to 1 the coefficient of consecutiveness is, the more focus the security team should put on the computer. However, if a computer only appears once in the IPS dataset, it means that *Week* = 1. Nonetheless, it does not necessarily mean that the security team should focus on that computer. This emphasizes that the number of weeks is also an important metric.

4.1.2 Number of Weeks

The number of weeks for which at least one alert was associated to the computer is the second metric.

However, the case where the number of weeks is 1 may be misleading. In this case (the computer was recorded as an attacker only for one week along the considered period of time), the coefficient of consecutiveness would be 1 and the computer would be reported to the security team. Considering the computers for which *Week* = 1 would raise a lot of alerts for computers that are in fact not corrupted. Therefore, we discard for the study all computers where week = 1.

Hence, the minimum is *Week* = 2 and the maximum is the number of weeks during which data have been collected.

The number of weeks reflects the frequency at which the computer appears in the IPS dataset. A computer with a large number of weeks reveals that the computer is potentially corrupted and has not been checked.

4.1.3 Number of Signatures

Finally, we believe that the number of distinct attack signatures associated with a given computer is important. It reflects the range of different attack types one computer seemed to have launched. Note that a great number of distinct signatures might also reveal that the computer contains several vulnerabilities.

The minimum number is 1 and the maximum is the total number of existing distinct signatures in the IPS.

4.2 Level of Criticality

We define the level of criticality of a computer as the 3-tuple {*Cons*, *Week*, *Sign*} (*Cons* stands for the coefficient of consecutiveness, *Week* for the number of weeks, *Sign* for the number of signatures). The higher the level of criticality, the more important it is for the security team to check that computer.

We identify three levels of interest: high concern, concern, and low concern. We define thresholds for each metric so that the interval is cut into three intervals: *C1* and *C2* are thresholds for the consecutiveness factor, *W1* and *W2* for the number of weeks, *S1* and *S2* for the number of distinct signatures. We decided to visualize each computer by using a Cartesian coordinate system: the coordinates are the consecutiveness factor, the number of weeks and the number of signatures. In other words, each computer is represented in a 3-D space. Hence, by considering the thresholds and the 3-D space, we can visualize a cube that is cut into 27 sub-cubes (Figure 2a).

We then introduce three colors associated with the three levels of criticality: 1) green regions (G) depict computers of low concern, 2) yellow regions (Y) group computers of concern that should be checked by the security team, and 3) red regions (R) show computers of high concern that should be addressed in priority. For each sub-cube, a security expert helped us decide on their level of criticality and thus their associated color. Figure 2b depicts the colors selected for the 27 sub-cubes.

4.3 Method for Identifying Computers of Concern

The method consists in five steps: 1) analysis of the IPS dataset to identify computers that were the source of alerts, 2) calculation of the level of criticality for each identified computer, 3) determination of thresholds for the three metrics, 4) investigation of computers in the red region and 5) investigation of computers in the yellow region.

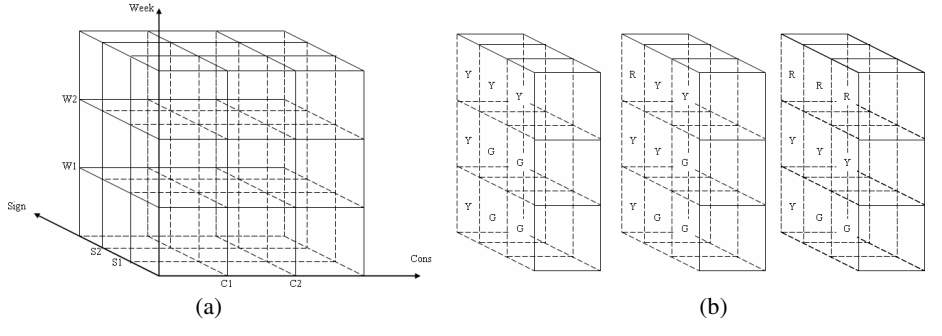

Fig. 2. Visualization of Metrics (a) and Colored Zones (b)

Step 1: Analysis of the IPS dataset
The identification of computers that were the source of alerts in the IPS dataset is done through the extraction of the internal IP addresses that appear as the source of alerts in the IPS dataset.

Step 2: Calculation of the level of criticality
To calculate a level of criticality, a period of time for which to calculate the metrics needs to be defined. We advise selecting a period long enough to allow a metric like the

coefficient of consecutiveness to be relevant (at least 5 weeks for the coefficient of consecutiveness to be meaningful).

Step 3: Determination of thresholds for the three metrics
We believe that threshold values (*C1*, *C2*, *W1*, *W2*, *S1*, *S2*) are organization dependent. Characteristics, such as the size of the organization, the type of the organization can greatly differ between organizations. In that sense, we advise each organization to choose its own thresholds.

Steps 4 and 5: Investigation of computers in the red and yellow regions
As the method consists in ranking computers in function of the level of criticality in order to focus on the computers of main concern, the security team should focus in priority on the computers in the red region.

Depending on the available sources of information, checking a potentially corrupted computer would include:

- Using the IPS dataset to look at the date and time of events,
- Using the IPS dataset to understand the attack type,
- Investigating previous incidents with that particular IP address.

The method tends to identify computers that appear frequently in the IPS dataset: those are the computers in the red and yellow regions (the frequency is reflected by the metrics *Cons* and *Week, Sign* interferes in making the distinction between the red and yellow regions). Hence, our method will not raise a flag for a computer that is involved in a single alert that could be harmful. Therefore, the method does not identify all potential corrupted computers.

Also, the method identifies computers that may be corrupted or not. For the remaining of the paper, we call:

- False negatives: corrupted computers that have not been identified by the method,
- True positives: corrupted computers that have been identified by the method,
- False positives: non-corrupted computers that have been identified by the method,
- True negatives: non-corrupted computers that have not been identified by the method.

The thresholds *C1*, *C2*, *W1*, *W2*, *S1* and *S2* are chosen by making a trade-off between the number of true positives and the number of false negatives.

5 Evaluation

5.1 Approach

In this section, we will evaluate the presented method. We will study IPS event data collected on a large public university (University of Maryland) composed of about 40,000 computers. The considered IPS dataset covers a period of 17 months, from

September 1st 2006 to January 31st 2008. The IPS raised an average of around 142 alerts per day during the studied period for computers inside the campus that are detected transmitting potentially malicious traffic toward computers outside the campus. Over the 17 months, 1,441 different computers inside the organization that launched at least one attack were identified.

First, we need to define a time period on which to apply the metrics. The campus is much less populated during the summer break (3 months) and the winter break (1 month). In other words, the traffic recorded by the IPS may drop during these periods due to fewer students/computers. In order not to bias the results, we should apply the metrics over a period greater than 3 months. We decided to apply the metrics over a 6-month period. In order to show how the metrics evolved over time, we calculated the metrics for increments of 2 weeks. On each period of 6 months, we extracted a list of computers and calculated the associated metrics.

We then asked the Director of Security of the Office of Information Technology at the University of Maryland, to indicate which computers were corrupted among the ones identified by our method. To do so, the Director of Security needed to investigate every computer. This step relies on expert judgment and human activity, as opposed to an automated investigation. As previously stated, we believe that computers for which *Week* = 1 are of less interest that those that appear at least two weeks over a 6-month period. By eliminating those computers, we are left with 303 computers to investigate.

We recognize that we rely on expert judgment to indicate which computers are corrupted. Another security expert might provide slightly different results. To decrease the potential bias due to expert judgment, we asked the Director of Security: 1) to use a systematic method for deciding if a computer is corrupted, and 2) to be conservative in his judgment (the Director will declare a computer corrupted (respectively non-corrupted) only if he is sure that the computer is corrupted (respectively non-corrupted)). Such requirements led to many investigated computers without clear decision. Among the 303 investigated computers, for 76 (25%) of them it was unclear whether they were corrupted. One reason is that the analyzed data went back to September 2006 making it difficult to make sure if the flagged computer was indeed corrupted.

First, in order to investigate the computers to determine if they are corrupted, the Director of Security needed the following information:

- For each computer: the number of alerts triggered in the IPS, the signature list associated to these alerts (SL), the time span for these alerts by signature, the list of computers targeted (target list TL), the list of incidents associated to the computer,
- A list of signatures known to trigger false alarms,
- A list of signatures known to be non-malicious.

Figure 3 depicts the sequential questions to answer regarding a given computer to determine if it is corrupted (C), non-corrupted (NC), or undetermined (O for other). If the answer to a question is “yes”, the computer can be classified and the Director of Security investigates another computer. If the answer is “no”, the Director of Security moves to the next question. These steps are the ones that were followed by the Director of Security to investigate the computers in order to evaluate the suggested method.

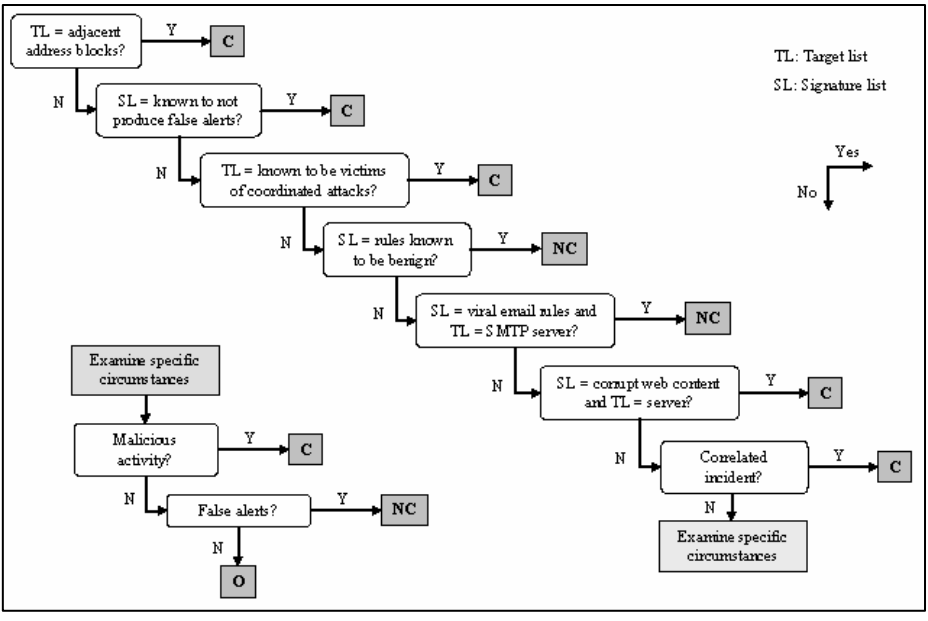

Fig. 3. Flowchart of the Steps of the Investigation

First, classifying computers as (non)-corrupted involves investigating the target list (TL): does the target list contain computers in the adjacent address blocks? If yes, it is possible that the computer is scanning the adjacent IP addresses range in order to detect computers. In that case, the computer is classified as corrupted. Otherwise, the signature list (SL) needs to be investigated: does the signature list contain signatures known to not produce false alerts? If yes, the computer is classified as corrupted. Six sequential steps consist in investigating the target list and the signature list. The seventh step aims at searching into the incident data in order to find an incident report involving the computer under investigation: if there is an incident report to support the alert associated with the computer, then the computer is classified as corrupted. If none of these steps allows classifying the computer as (non)-corrupted, the Director of Security will examine the specific circumstances of the alerts: if the investigation reveals malicious activity, the computer is corrupted; otherwise, if the investigation indicates false alerts, the computer is non-corrupted; otherwise, the computer is classified as undetermined.

Out of 303 investigated computers, 91 (30%) were identified as corrupted and 136 (45%) were identified as non-corrupted. One important measure is to find how many among the 303 identified computers led to an interesting investigation (independently of the outcome). The issue is whether the method identifies computers worth investigating or flags computers clearly of no concern leading to a waste of the time for the security team. Among the 303 flagged computers, the Director of Security found that the investigation was useful for all the identified computers. Indeed, either the computer is declared corrupted and the security team did check it or should have checked it, or the computer is not corrupted and the IPS itself needs to be retuned to reduce the number of alerts raised for non-corrupted computers. This high percentage indicates

that the proposed method is already of practical use for the security team. Although the number of non-corrupted investigated computers is high, the non-corrupted computers may reveal events that could not have been identified otherwise. For example, we identified an event where 64 systems tried to access Facebook using a suspicious PHP argument and users who operated Nmap. The computers involved in these two events were identified as non-corrupted but provided an additional insight into the organization's security.

The next step is to assess our method to know if it correctly identifies the (non)-corrupted computers. Each computer will be assigned a color: red (R), yellow (Y), green (G), and an investigation result: corrupted (C) or non-corrupted (NC) computer. A computer that was in the red or yellow regions and was identified as corrupted is a true positive. On the contrary, a computer that was in the green region and was identified as corrupted is a false negative. All combinations of color and investigation result are given in Table 1. Note that when the investigation could not tell if a computer was corrupted or not, we will use O (O stands for "Other"). For example, RO groups computers that are in the red region and that could not be identified as corrupted or non-corrupted by the Director of Security.

Table 1. All Combinations of Color and Investigation Result

Color	Investigation result	Notation	Conclusion
R	C	RC	True Positive (TP)
R	NC	RNC	False Positive (FP)
Y	C	YC	True Positive (TP)
Y	NC	YNC	False Positive (FP)
G	C	GC	False Negative (FN)
G	NC	GNC	True Negative (TN)
R	O	RO	-
Y	O	YO	-
G	O	GO	-

5.2 Results

We studied 23 periods of 6 months from September 1st 2006 to January 31st 2008 with increments of 2 weeks. Period 1 is the period from September 1st 2006 to February 28th 2007. Period 2 covers September 15th 2006 to March 14th 2007, etc. Period 23 defines the period from August 1st 2007 to January 31st 2008. For each period, we extracted the address of the organization computers that raised at least one alert corresponding to an attack towards a computer outside the University of Maryland and calculated the associated metrics. We applied the following thresholds: $C1 = 0.5$ and $C2 = 0.8$ for the coefficient of consecutiveness, $W1 = 2$ and $W2 = 4$ for the number of distinct weeks, $S1 = 1$ and $S2 = 2$ for the number of distinct signatures. For each of the 23 periods of 6 months, our method automatically puts each flagged computers in a green, yellow or red region. According to the identification of the (non)-corrupted computers by the Director of Security, we can calculate 1) the number of true/false positives based on the results in the yellow and red regions, and 2) the number of true/false negatives based on the results in the green region. The results are shown in Table 2.

Note first that the number of computers for which it could not be decided whether they were corrupted or not highly depends on the region. In the red region, they represent 12% (Period 1), 0% (Period 12) and 20% (Period 23). In the yellow region, we find 26% (Period 1), 38% (Period 12), and 36% (Period 23). In the green region, we have 71% (Period 1), 54% (Period 12), and 32% (Period 23). It is interesting to note that often the red region has the lowest percentage and the green region has the highest percentage of computers that could not be clearly identified as (non)-corrupted. This increases the confidence in our method since the computers in the red region should have the highest likelihood of being corrupted and the green region should have a much lower likelihood of being corrupted. This shows that the information provided to the security team should be useful as it seems to rank the computers based on the likelihood of corruption.

Graphs of the evolution of true positives, false positives, true negatives and false negatives over the 23 periods are shown in Figure 4. The results show that the method is improving regarding the number of true negatives. At Period 1, among the computers in the green region (i.e., computer of low concern), only 10% were not corrupted. However, the trend significantly changes over time. At Period 23, among the computers in the green region, 91.7% were not corrupted.

Table 2. Results of the Evaluation

Period	RM	RNM	YM	YNM	GM	GNM	OR	OY	OG	TP	FP	TN	FN	TP/P (%)	FP/P (%)	TN/N (%)	FN/N (%)
1	19	3	24	10	18	2	3	12	49	43	13	2	18	76.8	23.2	10.0	90.0
2	19	4	30	10	17	1	3	13	46	49	14	1	17	77.8	22.2	5.6	94.4
3	21	5	29	8	17	1	3	14	44	50	13	1	17	79.4	20.6	5.6	94.4
4	17	5	34	7	19	1	2	16	39	51	12	1	19	81.0	19.0	5.0	95.0
5	14	5	35	7	17	1	2	15	38	49	12	1	17	80.3	19.7	5.6	94.4
6	13	5	33	6	20	8	2	15	35	46	11	8	20	80.7	19.3	28.6	71.4
7	13	5	30	5	20	14	2	11	40	43	10	14	20	81.1	18.9	41.2	58.8
8	10	4	35	4	18	14	1	13	34	45	8	14	18	84.9	15.1	43.8	56.2
9	10	3	33	5	17	13	2	12	36	43	8	13	17	84.3	15.7	43.3	56.7
10	11	3	29	3	17	11	2	11	39	40	6	11	17	87.0	13.0	39.3	60.7
11	10	2	24	4	17	12	2	8	38	34	6	12	17	85.0	15.0	41.4	58.6
12	12	2	13	5	16	12	0	11	33	25	7	12	16	78.1	21.9	42.9	57.1
13	9	1	13	6	13	12	0	10	34	22	7	12	13	75.9	24.1	48.0	52.0
14	6	0	11	6	13	18	0	6	30	17	6	18	13	73.9	26.1	58.1	41.9
15	3	1	10	7	8	19	0	5	30	13	8	19	8	61.9	38.1	70.4	29.6
16	3	0	6	8	6	20	0	7	30	9	8	20	6	52.9	47.1	76.9	23.1
17	3	2	4	9	5	39	0	6	30	7	11	39	5	38.9	61.1	88.6	11.4
18	3	2	3	10	4	36	0	6	27	6	12	36	4	33.3	66.7	90.0	10.0
19	2	3	3	8	5	39	0	6	28	5	11	39	5	31.2	68.8	88.6	11.4
20	2	3	1	8	6	41	0	6	28	3	11	41	6	21.4	78.6	87.2	12.8
21	2	3	0	8	6	43	0	5	28	2	11	43	6	15.4	84.6	87.8	12.2
22	2	3	0	8	6	44	0	5	26	2	11	44	6	15.4	84.6	88.0	12.0
23	1	3	1	6	4	44	1	4	23	2	9	44	4	18.2	81.8	91.7	8.3

On the other hand, the method identifies a high percentage of true positives at Period 1 but a low percentage at Period 23. At Period 1, the method identified 76.8% of the computers in the red and yellow regions as being indeed corrupted. At Period 23, the

method only found 18.2% of corrupted computers in the red and yellow regions. These numbers might indicate that our method is worsening over time. More details are necessary to better understand the reasons for the obtained results. As expected, over time, the security team learned how to integrate the results provided by the IPS in their overall security solution. The number of identified corrupted computers is 61 at Period 1, 41 at Period 12, and only 6 at Period 23. This clearly indicates that the IPS is helping the security team improving the overall organization's security. These numbers help putting in perspective the only 18.2% of corrupted computers in the regions of concern. At Period 23, only 5 computers were placed in the red region and 11 in the yellow region. Among them, 3 computers were incorrectly put in the red region when they were not corrupted and 6 in the yellow region. On the other hand, at Period 23, most computers were placed in the green region (71). Among them, only 4 (5.6%) were incorrectly put in the green region, i.e., they were identified as of low concern when in fact they were corrupted. The method seems thus to be able to correctly identify the biggest volumes of events, i.e. corrupted computers at Period 1 and non-corrupted computers at Period 23.

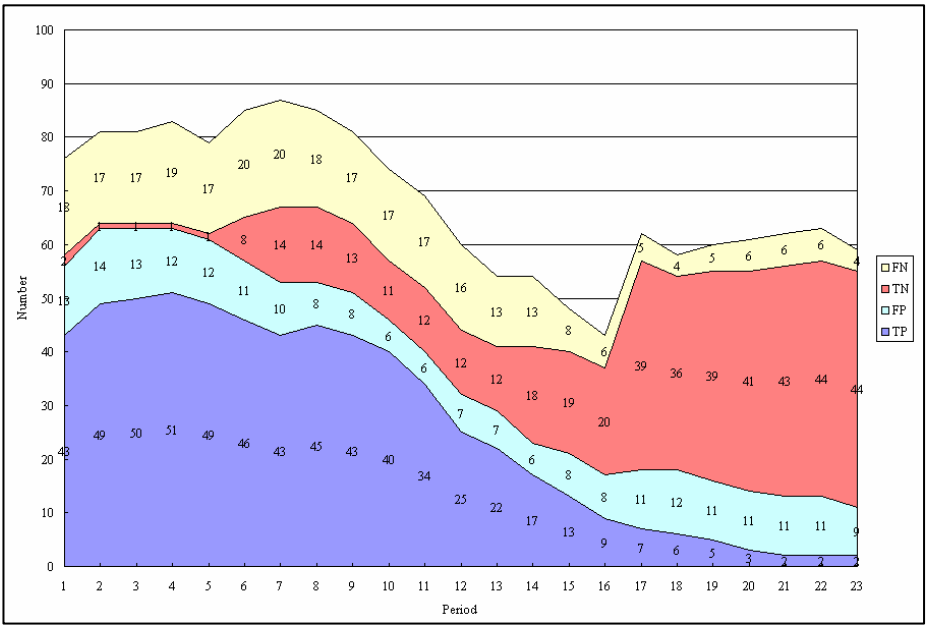

Fig. 4. Evolution of False Negatives (FN), True Negatives (TN), False Positives (FP) and True Positives (TP)

6 Conclusion

We presented a method to extract useful information from imperfect IPS event data in order to rank potentially corrupted computers in an organization. We introduced three metrics to quantify the level of criticality of a computer: the coefficient of consecutiveness, the number of distinct weeks and the number of distinct signatures. The

method classifies computers into regions of main concern (red regions), concern (yellow region), or lower concern (green region). We applied the method to IPS event data collected in an organization of about 40,000 computers. We evaluated our method by comparing the results obtained by our method with the identification of (non)-corrupted computers by a security expert. We showed that: 1) the percentage of computers identified as corrupted is higher for computers in the red region than for computers in the green region, 2) the trend of the number of true negatives increases over time, 3) the security team seems to integrate the IPS in their overall organization's security as the number of computers identified as corrupted decreases over time.

Acknowledgements

The authors would like to thank Robin Berthier for the fruitful discussions during the development of this paper.

This research has been supported in part by NSF CAREER award 0237493.

References

1. Bailey, M., Cooke, E., Jahanian, F., Provos, N., Rosaen, K., Watson, D.: Data Reduction for the Scalable Automated Analysis of Distributed Darknet Traffic. In: Proceedings of the USENIX/ACM Internet Measurement Conference, New Orleans (2005)
2. Sung, M., Haas, M., Xu, J.: Analysis of DoS attack traffic data. In: 2002 FIRST Conference, Hawaii (2002)
3. Viinikka, J., Debar, H., Mé, L., Séguier, R.: Time series modeling for IDS alert management. In: Proceedings of the 2006 ACM Symposium on Information, computer and communications security, pp. 102–113. ACM Press, New York (2006)
4. Clifton, C., Gengo, G.: Developing custom intrusion detection filters using data mining. In: MILCOM 2000. 21st Century Military Communications Conference Proceedings, vol. 1 (2000)
5. Cuppens, F.: Managing alerts in a multi-intrusion detection environment. In: Proceedings of the 17th Annual Computer Security Applications Conference, vol. 32. IEEE Computer Society, Los Alamitos (2001)
6. Cuppens, F., Miege, A.: Alert correlation in a cooperative intrusion detection framework. In: IEEE Symposium on Security and Privacy, pp. 202–215 (2002)
7. Julisch, K.: Mining Alarm Clusters to Improve Alarm Handling Efficiency. In: Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC), pp. 12–21 (2001)
8. Julisch, K.: Data mining for Intrusion Detection. Applications of Data Mining in Computer Security. Kluwer Academic Publishers, Dordrecht (2002)
9. Julisch, K., Dacier, M.: Mining intrusion detection alarms for actionable knowledge. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 366–375. ACM Press, New York (2002)
10. Manganaris, S., Christensen, M., Zerkle, D., Hermiz, K.: A data mining analysis of RTID alarms. Computer Networks 34(4), 571–577 (2000)
11. Pietraszek, T.: Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection. In: Recent Advances In Intrusion Detection: 7th International Symposium. Springer, Heidelberg (2004)

12. Debar, H., Wespi, A.: Aggregation and correlation of intrusion-detection alerts. Recent Advances in Intrusion Detection. Springer, Heidelberg (2001)
13. Morin, B., Me, L., Debar, H., Ducasse, M.: M2D2: A Formal Data Model for IDS Alert Correlation. In: Recent Advances in Intrusion Detection: 5th Internatonal Symposium. Springer, Heidelberg (2002)
14. Ning, P., Xu, D., Healey, C., Amant, R.S.: Building attack scenarios through integration of complementary alert correlation methods. In: Proceedings of the 11th Annual Network and Distributed System Security Symposium, pp. 97–111 (2004)
15. Valdes, A., Skinner, K.: Probabilistic Alert Correlation. In: Proceedings of the Fourth International Workshop on the Recent Advances in Intrusion Detection (2001)
16. Valeur, F., Vigna, G., Kruegel, C., Kemmerer, R.: Comprehensive approach to intrusion detection alert correlation. IEEE Transactions on Dependable and Secure Computing 1(3), 146–169 (2004)