Malware Characterization through Alert Pattern Discovery

Steven Cheung
Computer Science Laboratory
SRI International
Menlo Park, CA 94025

Alfonso Valdes
Computer Science Laboratory
SRI International
Menlo Park, CA 94025

Abstract

We present a novel alert correlation approach based on the factor analysis statistical technique for malware characterization. Our approach involves mechanically computing a set of abstract quantities, called *factors*, for expressing the intrusion detection system (IDS) alerts pertaining to malware instances. These factors correspond to patterns of alerts, and can be used to succinctly characterize malware. Unlike most existing alert correlation approaches for multistep attacks, our approach does not require predefined attack models for characterizing complex multistep attacks, and discovers potentially unknown relationships among alert types. Moreover, it requires relatively little alert information. As such, this approach is suitable for analysis pertaining to large-scale, privacy-preserving alert repositories.

Initial experimental results indicate that our approach is useful in facilitating automated IDS alert pattern discovery, and in characterizing malware that manifests as multiple attack steps. Also, it may be used in identifying redundant signatures, enabling IDS performance tuning. Specifically, we examined the Snort rule identifiers (SIDs) of the alerts generated by the BotHunter tool, developed in the Cyber-Threat Analytics project, considering which SIDs co-occur pertaining to the same identified bot instance. Our exploratory analysis indicates that IDS alerts corresponding to bots can be expressed in terms of a small number of factors. Also, some bot families have distinguishing factor patterns.

1 Introduction

The proliferation and diversification of propagating malware is rapidly outstripping the capacity of the research and network defense community to identify and classify outbreaks. To date, attempts to comprehend malware typically involve labor-intensive and time-consuming analysis to identify variants and families of malware according to the exploits used, command and control activities, and other characteristics. The analyst-intensive nature of these efforts gives malware a critical time window during which an attack may do widespread damage before it is adequately understood so that effective defensive measures can be deployed.

Honeynets address part of this difficulty by allowing researchers to harvest traffic strongly suspected of being malicious. However, much traffic to a honeynet may consist of unsuccessful infection attempts. Dialog-based correlation, a technique introduced in BotHunter [5], can reliably identify instances of successful bot infections by correlating IDS alerts pertaining to different phases of bot infections. To model the infection process, BotHunter detects five common types of (loosely ordered) communication flows between an internal host and a set of external hosts that may be observed during bot infections:

- E1: External-to-internal inbound scan
- E2: External-to-internal inbound exploit
- E3: Internal-to-external binary acquisition
- E4: Internal-to-external command and control communication
- E5: Internal-to-external outbound infection scanning

Events pertaining to these types have a score. Only when a subset of these events is detected with a total score exceeding a specified threshold, BotHunter will then report a bot infection. This detection approach can tolerate the absence of some dialog events, e.g., failed detections for E1 or E2 events because of insufficient IDS coverage.

This paper examines bot infection profiles collected via BotHunter on a high-interaction honeynet. We analyzed 106 infection instances, which triggered a total of 26 unique Snort rules in various combinations. We examined the triggered rules in the various instances using

the statistical technique of factor analysis (e.g., [11]), described in Section 3. To validate the discovered rule patterns, we harvested the corresponding malware binaries and submitted them to `www.virustotal.com` for labeling.

This paper presents the feasibility and utility of the factor analysis technique as applied to Snort SIDs. We believe that this approach may be applicable to other event types from a collection of heterogeneous sensors, such as alerts from a variety of IDSs and firewalls. The technique uses very little information, and in particular does not use the IP addresses of attack traffic. As such, it is suitable for analysis of large-scale, privacy-preserving alert repositories such as that of the Cyber-Threat Analytics (Cyber-TA) project.

The contributions of this paper are summarized as follows. We present a novel alert correlation approach based on the factor analysis statistical technique for characterizing malware. This approach identifies patterns in the IDS alerts triggered by malware, facilitating rapid and consistent identification and classification of existing and emerging malware threats. This is accomplished without predefined attack models, which are typically labor and time intensive to construct. Moreover, the method discovers potentially unknown relationships among event types. Our approach can also facilitates IDS performance tuning by uncovering redundant IDS rules, which exhibit as co-occurring alert IDs for malware instances, and are highly correlated with the same factors. We conducted an experiment to validate the usefulness of the approach. Our initial results are promising. For a set of bot instances we analyzed, we achieve a significant dimensionality reduction by using factor analysis: over 89% of the variance in the alerts (of 26 types) can be explained by five factors. Every malware instance in the dataset can be expressed succinctly by a few factors. To examine the consistency of our approach, we compare our results with the labels assigned to the binaries of the bot instances by existing malware detection products.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes our approach and reviews the factor analysis technique. Section 4 presents the results of performing factor analysis on the BotHunter alerts. We observed that five factors are sufficient to enable us to characterize the dataset containing 26 different Snort signatures. Moreover, there are only seven patterns of (significant) factor loadings. We are able to attribute some of these patterns to command and control activity, and others to specific malware species. Section 5 describes an attempt to validate the experimental results using malware labels obtained from third-party virus research and classification efforts. Section 6 discusses our findings. Section 7 concludes and presents our plan for future work.

2 Related Work

To better comprehend alerts generated by IDS and firewalls to achieve situation awareness, a significant amount of research has been performed on alert correlation.

By analyzing potentially a large volume of alerts (possibly from a myriad of heterogeneous sensors), identifying the security-critical ones and discounting the false alarms, and inferring the relationships among the alerts, alert correlation aims to generate high-level digests that summarize and explain the alerts.

Previous alert correlation work includes Goldman et al.'s Scyllarus correlation framework [4], Julisch's work using alarm clustering to perform root cause analysis [6], Ning et al.'s abstraction-based intrusion detection approach [8], Porras et al.'s M-Correlator [9], a mission-impact-based correlation system that uses common-attribute-based aggregation, network topology analysis, and mission criticality specification to perform alert fusion and ranking, Qin's and Lee's statistical causality analysis approach for alert correlation [10], Staniford et al.'s graph-based approach for detecting large-scale attacks [12], Valdes's and Skinner's probabilistic approach for correlating IDS alerts based on (possibly imperfect) matching of attributes [16], and Yang et al.'s distributed coordinated attack detection system called CARDS [19].

For detecting complex, multistep attacks, an attack modeling approach based on specifying the pre- and post-conditions of the individual attack steps has been proposed in Cuppens' and Ortalo's LAMBDA [3], Michel's and Mé's ADeLE [7], Ning et al.'s abstraction-based approach [8], and Templeton's and Levitt's Jigsaw [14]. Valeur et al. [17] proposed a state-transition-based modeling framework for detecting multistep attacks. To facilitate multistep attack modeling, Cheung et al. [2] presented attack patterns to ease reuse of attack module specifications. The effectiveness of these approaches strongly relies on the accuracy and the coverage of the attack models developed. Our work presents a complementary approach for detecting multistep attacks by discovering alert patterns without depending on predefined attack models.

Viinikka et al. [18] presented a time-series-based approach for modeling the regularities of and eliminating the background noise in the alert flows. Using that approach, security analysts can focus their resources on the more interesting alerts. Our work has some similarities with [18] in that both approaches attempt to extract patterns from the alert stream using event IDs. A main difference is that our approach focuses on patterns across different event IDs, whereas [18] focuses on the trends for individual event IDs.

Bailey et al. [1] examined the weaknesses of existing

antivirus products with respect to their consistency, completeness, and conciseness in classifying malware, and presented an automated malware classification scheme based on their behavior, such as file accesses and process creation, to address those issues. Our work shares some of the objectives of [1], but is based on correlating network IDS alerts using the factor analysis statistical technique.

3 Our Approach

Our approach is motivated by the observation that some bot profiles (i.e., the sets of IDS alerts generated for individual bot instances) have significant overlap, but are not necessarily identical. Moreover, some bot profiles exhibit distinguishing characteristics.

We hypothesized that different instances of a bot or different bot variants belonging to the same family may exhibit similar observable behavior from the network IDS viewpoint. Moreover, we hypothesized that different bot species or families have distinguishing network IDS alert profiles, as they may exploit different vulnerabilities, use different means to coordinate with command-and-control servers and other bots, and employ different techniques to propagate.

Factor analysis (and the closely related principal component analysis) are statistical techniques that may reduce a complex dataset containing measurements of a number of quantities to one with a lower dimension. These factors (or components) correspond to some commonalities among the quantities, and can sometimes reveal simplified, high-level structures that may not be observed directly.

A main objective of this study is to test our hypothesis through the use of a small set of factors that can be used to account for (most of) the variability of the observed IDS alerts. These factors, each corresponding to a set of alert IDs, may be used to succinctly characterize the malware instances. In other words, we can express the set of IDS alerts for each malware instance in terms of a small number of factor patterns.

To illustrate factor analysis, let us consider an example from [13]. In the example, we randomly selected a set of people from the population, and measured the sizes of different parts of their bodies, e.g., height, leg, waist, fingers. We would expect that many of the measurements would be correlated, and could be explained by an underlying common factor of body size, which is a high-level quantity not directly measured. Using the body size factor alone may not be sufficient to account for all (or most) of the variability of the measurements, and we may need additional factors such as lankiness. Factor analysis enables us to achieve economy of description (or dimensionality reduction), describing the measurements using a smaller number of components, without losing too much information.

In this paper, we gathered the sets of alert IDs, which are Snort rule identifiers in our experiment as described in Section 4, pertaining to malware instances. We converted the malware profiles to a matrix, with rows corresponding to malware instances, and columns to Snort rule identifiers. For example, a malware instance may trigger the following set of alerts:

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 445
  (msg:"E2[rb] NETBIOS SMB-DS Session Setup NTMLSSP
   unicode asn1 overflow attempt";
   ...  sid:3003; rev:4;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 445
  (msg:"E2[rb] SHELLCODE x86 inc ebx NOOP";
   ... sid:99998; rev:2;)
alert ip $EXTERNAL_NET $SHELLCODE_PORTS ->
  $HOME_NET any
  (msg:"E2[rb] REGISTERED FREE SHELLCODE x86 inc
   ebx NOOP"; ... sid:1390; rev:5;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 445
  (msg: "E2[rb] BLEEDING-EDGE EXPLOIT MS04-007
   Kill-Bill ASN1 exploit attempt";
   ... sid:2001944; rev:3;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 445
  (msg:"E3[rb] BotHunter MALWARE executable upload";
   ... sid:3000006; rev:99;)
alert tcp $EXTERNAL_NET !20 -> $HOME_NET any
  (msg:"E3[rb] BLEEDING-EDGE Malware Windows
   executable sent from remote host";
   ... sid:2001683; rev:3;)
alert tcp $EXTERNAL_NET !20 -> $HOME_NET any
  (msg:"E3[rb] BotHunter Malware Windows executable
   (PE) sent from remote host";
   ... sid:5001684; rev:3;)
```

This malware profile is transformed to a row of the matrix with the entries in columns pertaining to SIDs 3003, 99998, 1390, 2001944, 3000006, 2001683, and 5001684 equal 1, and the rest of the entries equal 0.

Formally, factor analysis is a statistical technique that examines the covariance or correlation structure among a number of variables, and then extracts factors that explain most of the information in the matrix. Briefly, for a set of observed variables $X_1, X_2, \ldots X_n$ (which in our analysis correspond to counts of Snort SIDs for each bot infection instance), the common factors $F_1, F_2, \ldots F_m$, and the unique factors $U_1, U_2, \ldots U_m$, the variables may be expressed as linear functions of the factors

$$
\begin{aligned}
X_1 &= a_{11}F_1 + a_{12}F_2 + \ldots + a_{1m}Fm + a_1U_1 \\
X_2 &= a_{21}F_1 + a_{22}F_2 + \ldots + a_{2m}Fm + a_2U_2 \\
&\vdots \\
X_n &= a_{n1}F_1 + a_{n2}F_2 + \ldots + a_{nm}Fm + a_nU_n
\end{aligned}
$$

Factor analysis computes the coefficients a_{ij}. Factor analysis is similar to principal component analysis, a more widely used statistical technique. Unlike principal component analysis, factor analysis uses variable-unique factors; on the other hand, principal component analysis assumes that all the variances of the observed variables can be explained by a set of common factors. Also, factor analysis finds a set of factors such that the observed

variables have substantial loadings on as few factors as possible, which makes it easier to interpret the results.

The factor coefficients are referred to as *factor loadings*, and these are generally scaled so that the vector of factor loadings has unit L2 norm. Factor analysis is frequently used as a dimensionality reduction technique, where the most significant factors are selected based on the fraction of the total variance explained. The loading on any individual variable can be positive or negative, and the absolute value is indicative of the importance of the underlying variable in the respective factors. Readers are referred to [13] and [11] for more information about the factor analysis technique.

4 Experimental Results

We used a dataset from the Cyber-TA malware analysis Webpage (`www.cyber-ta.org/releases/malware-analysis/public/`). The alerts were collected by a Snort IDS deployed in a honeynet. This sensor uses a custom Snort ruleset for detecting bot-specific activities. The IDS alerts generated by the Snort sensor over a 24-hour period were partitioned into 106 sets, corresponding to different malware instances. Moreover, the dataset contains 26 different Snort IDs.

The data records for the factor analysis consist of the counts of each of the SIDs per infection instance. We observed that in most cases, this was 1 or 0. We have performed factor analysis that considers rule incidence rather than count—in other words, the data record would be binary, with a 1 indicating that the corresponding SID was triggered in the infection instance. The result for this "binary" dataset is essentially the same as that of the original dataset.

Visual examination of the correlation matrix revealed interesting structure. First, we observed that some SID pairs have a correlation of 1.0, indicating redundancy and the potential for rule pruning for performance reasons. While it may be the case that perfectly correlated SIDs in our data are not perfectly correlated in general, we are limiting our analysis to likely successful infection instances as identified by dialog-based correlation. Thus, this observed perfect correlation appears to hold in some interesting cases. SID subsets for which we observed perfect correlation are $\{1390, 2001944, 3000006, 99998\}$, $\{1444, 3001441\}$, $\{2000046, 99906\}$, $\{2000047, 2001056\}$, and $\{2001184, 2002029\}$. For some of these subsets, potentially one could select the maximum coverage rule, and disable the others as redundant.[1]

We used Matlab to perform factor analysis [15]. The input of factor analysis is an $n \times p$ data matrix—where n is the number of measurements and p is the number of variables—and generates a set of factors and the corresponding loadings and variance.

We used the latent root criterion[2] for selecting factors, which turned out to be five. After applying the varimax factor rotation, we obtained the factor loading table as shown in Table 1. In the table, each row corresponds to an event ID. For our analysis, these correspond to Snort SIDs, but this need not be the case in general. Conceptually, the approach is applicable to a heterogeneous sensor environment, containing events from multiple sensors.

The first column in the table corresponds to SIDs. Columns 2 through 6 represent the loadings for the selected factors, respectively. The last column corresponds to the communality (i.e., the fraction of the total variance captured by the selected factors). The larger the communality value is (i.e., closer to 1), the more variability of the corresponding variable (or SID) is accounted for using the common factors. On the other hand, a small communality value (i.e., closer to 0) means that the common factors cannot effectively explain the variability of the corresponding variable.

The results indicate that all observed variables have significant loadings on one or two factors. Moreover, the factor loadings often correspond to distinct patterns over the variables in question, which can be used to group the variables. Based on the factor loadings, we can form the following groups.

Group 1

Five Snort IDs have substantial negative loadings for the first factor.

```
1390: alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any
(msg:"E2[rb] REGISTERED FREE SHELLCODE x86 inc ebx NOOP"; ...

2001944: alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(msg: "E2[rb] BLEEDING-EDGE EXPLOIT MS04-007 Kill-Bill ASN1 exploit
 attempt"; ...

3000006: alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(msg:"E3[rb]  BotHunter MALWARE executable upload"; ...

3003: alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(msg: "E2[rb] NETBIOS SMB-DS Session Setup NTMLSSP unicode asn1
 overflow attempt"; ...

99998: alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(msg:"E2[rb] SHELLCODE x86 inc ebx NOOP"; ...
```

Group 2

Seven Snort IDs have substantial positive loadings for the first factor.

```
2000032: alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(msg: "E2[rb] BLEEDING-EDGE EXPLOIT LSA exploit"; ...

2000033: alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(msg: "E2[rb] BLEEDING-EDGE EXPLOIT MS04011 Lsasrv.dll RPC
 exploit (WinXP)"; ...

2001569: alert tcp $HOME_NET any -> any 445
(msg: "E5[rb] BLEEDING-EDGE Behavioral Unusual Port 445 traffic,
 Potential Scan or Infection"; ...

2466: alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(msg:"E2[rb] NETBIOS SMB-DS IPC$ unicode share access"; ...

3000000: alert tcp $EXTERNAL_NET any -> $HOME_NET 1030:1040
(msg: "E3[rb] BotHunter HTTP-based .exe Upload on backdoor port"; ...

3000003: alert tcp  $HOME_NET 1028:1040 -> $EXTERNAL_NET any
```

SID	Fac. 1	Fac. 2	Fac. 3	Fac. 4	Fac. 5	Comm.
1390	-0.9629	-0.0113	-0.1155	-0.1942	0.1184	0.9924
2001944	-0.9629	-0.0113	-0.1155	-0.1942	0.1184	0.9924
3000006	-0.9629	-0.0113	-0.1155	-0.1942	0.1184	0.9924
3003	-0.9477	-0.0167	-0.1129	-0.1937	0.1172	0.9625
99998	-0.9629	-0.0113	-0.1155	-0.1942	0.1184	0.9924
2000032	0.9225	-0.0042	-0.2663	0.2118	0.1329	0.9845
2000033	0.8993	0.0468	-0.2473	-0.2723	0.1255	0.9620
2001569	0.7990	0.0256	0.1626	-0.2458	0.1692	0.7545
2466	0.9550	0.0067	0.1120	0.2026	0.1557	0.9898
3000000	0.7969	0.1429	-0.2557	-0.2864	0.1383	0.8220
3000003	0.7502	0.1266	-0.2113	-0.2943	0.1379	0.7291
99913	0.9248	0.0622	0.1313	-0.2938	-0.1231	0.9780
2000345	-0.1712	-0.8574	-0.0251	-0.0484	0.0279	0.7682
2001184	0.1046	-0.8481	0.0033	0.2054	-0.0070	0.7725
2002024	-0.2044	-0.8220	-0.0299	-0.0602	0.0332	0.7231
2002025	-0.0909	-0.8936	-0.0157	0.0035	0.0153	0.8072
2002028	0.0217	-0.8904	-0.0049	0.1095	0.0004	0.8053
2002029	0.1046	-0.8481	0.0033	0.2054	-0.0070	0.7725
2000047	0.0595	0.0284	0.9852	-0.0294	0.0554	0.9789
2001056	0.0595	0.0284	0.9852	-0.0294	0.0554	0.9789
2000046	0.0397	-0.1026	-0.0361	0.9724	0.0139	0.9591
3000004	0.1059	-0.2377	0.5587	0.7425	0.0379	0.9326
99906	0.0397	-0.1026	-0.0361	0.9724	0.0139	0.9591
1444	0.0032	0.0164	0.0096	-0.0357	-0.9895	0.9808
3001441	0.0032	0.0164	0.0096	-0.0357	-0.9895	0.9808
2001683	0.0491	-0.0984	-0.6592	-0.1873	0.4522	0.6863

Table 1: Output of Factor Analysis

```
(msg: "E3[rb] BotHunter HTTP-based .exe Upload on backdoor port"; ...

99913: alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(msg:"E2[rb] SHELLCODE x86 0x90 unicode NOOP"; ...
```

Group 3

Six Snort IDs have substantial negative loadings for the second factor.

```
2000345: alert tcp $HOME_NET any -> $EXTERNAL_NET !6661:6668
(msg: "E4[rb] BLEEDING-EDGE ATTACK RESPONSE IRC - Nick change on
 non-std port"; ...

2001184: alert tcp $EXTERNAL_NET any -> $HOME_NET any
(msg: "E4[rb] BLEEDING-EDGE WORM RXBOT / RBOT Vulnerability Scan"; ...

2002024: alert tcp $HOME_NET any -> $EXTERNAL_NET !25
(msg: "E4[rb] BLEEDING-EDGE TROJAN IRC NICK command"; ...

2002025: alert tcp $HOME_NET any -> $EXTERNAL_NET any
(msg: "E4[rb] BLEEDING-EDGE TROJAN IRC JOIN command"; ...

2002028: alert tcp $HOME_NET any -> $EXTERNAL_NET any
(msg: "E4[rb] BLEEDING-EDGE TROJAN IRC PONG response"; ...

2002029: alert tcp $EXTERNAL_NET any -> $HOME_NET any
(msg: "E4[rb] BLEEDING-EDGE TROJAN BOT - channel topic
 scan/exploit command"; ...
```

Group 4

Two Snort IDs have substantial positive loadings for the third factor.

```
2000047: alert tcp $EXTERNAL_NET any -> $HOME_NET 9996
(msg: "E3[rb] BLEEDING-EDGE VIRUS Sasser Transfer _up.exe"; ...

2001056: alert tcp $EXTERNAL_NET any -> $HOME_NET any
(msg: "E2[rb] BLEEDING-EDGE VIRUS W32/Sasser.worm.b -NAI-)"; ...
```

Group 5

Three Snort IDs have substantial positive loadings for the fourth factor.

```
2000046: alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(msg: "E2[rb] BLEEDING-EDGE EXPLOIT MS04011 Lsasrv.dll RPC
 exploit (Win2k)"; ...

3000004: alert tcp  $HOME_NET any -> $EXTERNAL_NET any
(msg: "E3[rb] BotHunter Scrip-based Windows egg download .exe"; ...

99906: alert tcp $EXTERNAL_NET any -> $HOME_NET 445
(msg:"E2[rb] SHELLCODE x86 0x90 unicode NOOP"; ...
```

Group 6

Two Snort IDs have substantial negative loadings for the fifth factor.

```
1444: alert udp $HOME_NET any -> $EXTERNAL_NET 69
(msg:"E3[rb] TFTP GET from external source"; ...

3001441: alert udp $HOME_NET any -> $EXTERNAL_NET 69
(msg:"E3[rb] TFTP GET .exe from external source"; ...
```

Group 7

Snort ID 2001683 has moderate loadings on the third and fifth factors.

```
2001683: alert tcp $EXTERNAL_NET !20 -> $HOME_NET any
(msg:"E3[rb] BLEEDING-EDGE Malware Windows executable sent from
 remote host"; ...
```

We used the groups to characterize malware instances based on the IDS alerts generated for them. For example, a bot profile (shown in Section 3) containing alerts with the SIDs 3330, 99998, 1390, 2001944, 3000006, and 2001683 may be characterized using Groups 1 and 7. In the next section, we will evaluate the grouping results obtained in this experiment.

5 Validation

The malware binaries captured by the Cyber-TA honeynet were submitted to `www.virustotal.com` for testing against a suite of malware detection products. We

Malware Label	Factor Group ID (# SID triggered/Total # SIDs)
Worm/Sasser.C	2(3/7),4(2/2),5(1/3)
Worm/Sasser.A.14	2(3/7),4(2/2),5(1/3)
Worm/Korgo.AE	2(7/7),7(1/1)
Worm/Korgo.AF	2(7/7),7(1/1)
Worm/Korgo.G.1	2(5/7),7(1/1)
Worm/Korgo.I	2(5/7),7(1/1)
Worm/Korgo.X	2(7/7),7(1/1)
Worm/Korgo.U	2(7/7),7(1/1)
Worm/Padobot.AA	2(5/7),7(1/1)
Worm/Padobot.P	2(7/7),7(1/1)
Worm/Padobot.Z.2	2(5/7),7(1/1)
∗W32.Spybot.Worm	1(5/5),3(2/6),7(1/1)
∗W32.Spybot.Worm	1(4/5),7(1/1)
∗W32.Spybot.Worm	1(5/5),7(1/1)
Worm/Rbot.328262	1(5/5),7(1/1)
Worm/Rbot.328262	2(2/7),5(3/3),7(1/1)
Worm/Rbot.545792.4	1(5/5),7(1/1)
Worm/Rbot.550912.1	1(5/5),7(1/1)
Worm/Rbot.328262	1(5/5),3(4/6),7(1/1)
Worm/Sdbot.557056.4	1(5/5),7(1/1)
Worm/Sdbot.557056.4	1(5/5),3(3/6),7(1/1)
Worm/SdBo.100864.22	2(4/7),3(6/6),5(1/3),7(1/1)
Worm/SdBo.100864.22	2(3/7),3(6/6),5(3/3),7(1/1)
Worm/SdBo.100864.22	2(2/7),5(3/3),7(1/1)
Worm/Rand	2(7/7),7(1/1)
W32/Virut.AF	2(6/7),7(1/1)
W32/Virut.AM	2(7/7),7(1/1)
W32/Virut.AO	1(5/5),7(1/1)
W32/Virut.AO	1(5/5),3(3/6),7(1/1)
W32/Virut.Gen	2(7/7),7(1/1)
?	1(5/5)
?	1(5/5),7(1/1)
?	2(1/7),6(2/2)
?	2(2/7),5(3/3)
?	2(2/7),5(3/3),7(1/1)
?	2(4/7),5(1/3),7(1/1)

Table 2: Mapping: Malware Labels and Group IDs

use the labels obtained from these products to further validate the factor analysis results. The property we want to validate is the consistency of the mapping between the factor patterns and the malware labels. Ideally, there is a one-to-one mapping between them, although that may not be possible in practice, thanks to the set of observable events (which in turn depends on the Snort ruleset employed) and the limitations of our approach.

The dataset has 106 malware instances. For each instance, we obtain the malware label specified by AntiVir (and by Symantec when AntiVir cannot provide a specific malware label[3]) and identify the corresponding alert patterns in terms of the above-mentioned groups.

Table 2 shows all distinct mappings between the malware labels and the groups identified by factor analysis for the malware instances. For example, the malware instance corresponding to the label "Worm/Sasser.C" triggered three out of seven rules in Group 2, both of two rules in Group 4, and one out of three rules in Group 5. The malware labels in the table were provided by AntiVir, except that those with a * prefix were provided by Symantec. The question marks correspond to malware instances that cannot be labeled by AntiVir and Symantec.

6 Discussion

Our experimental results show that the malware instances in our dataset can be explained using a small number of factors. These factors may or may not have human-level semantics; they are computed mechanically to reflect which Snort IDs tend to co-occur.

Let us examine the Snort ID groupings induced by the factors to attempt to understand what they mean from the intrusion detection viewpoint. Group 1 consists of rules for detecting external to internal infection, and one for downloading malware. From the correlation matrix among the SIDs of Group 1, they all have very large (pairwise) correlation coefficients (of 0.9 or above), as we expected. Also, we found that two rules in Group 1 have almost identical detection coverage (SIDs 1390 and 99998), except that they differ in their source/destination port specification. Group 2 is the largest group, consisting of seven SIDs. They correspond to Snort rules for detecting several phases of bot infection: exploit, binary acquisition, and internal-to-external outbound infection scanning. Group 3 corresponds to rules for detecting command-and-control channel activities, e.g., IRC transactions. Group 4 contains two rules for detecting the Sasser worm. Moreover, Group 6 contains two rules for detecting TFTP GET activities, possibly for downloading malware from external sources. These two rules have virtually identical detection coverage. Thus, one of them may be removed, for performance purposes.

From Table 2, we observe that malware instances belonging to the same family generally share similar alert patterns, although there are exceptions. For example, variants of the Korgo family have similiar factor pattern characterization. Moreover, the Sasser family has a distinguishing pattern, which involves Group 4 alerts.

Examination of the rules in Groups 1 and 6 reveals that some IDS rules are virtually identical in detection coverage. Thus, some of them can be removed from the rulebase of the IDS to improve its runtime performance without reducing its detection capabilities. For most IDS deployments, the number of rules is generally quite large (e.g., in the order of 1000's) and it is resource intensive to manually find redundant rules. Our approach facilitates the identification of redundant IDS rules based on runtime behavior.

We note that the alert patterns for W32/Virut.AM and W32/Virut.AO appear to be quite different. We hypothesize that it may partly depend on the classification schemes used by the antivirus companies. For instance, the malware instances labeled as W32/Virut.AM by AntiVir were labeled as W32.Korgo.S by Symantec, and the factor pattern for these malware instances resembles those for the instances labeled as Korgo.

Another interesting experimental result pertains to

malware instances that cannot be labeled by the existing antivirus products. As shown in Table 2, there are four such samples. Based on the similarities and differences between the factor patterns for the unlabeled bot instances and those for the labeled ones, one may infer new bot varieties or common lineage between the labeled ones and the unlabeled ones. For example, the factor pattern of the unlabeled sample involving Group 6 SIDs is quite different from those of the labeled ones, and could correspond to a new bot or to a major variant of the known bots. The factor group pattern corresponding to the fifth unlabeled malware instances is similar to those of Rbot.328262 and SdBo.100864.22, and thus they could be related malware species.

Not all of the experimental results are positive. We have noticed a few limitations or weaknesses of our approach:

- We observed an anomaly for using the identified common factors to characterize Worm/Rbot.328262—more than one malware instance with that same label have different alert patterns. We have two hypotheses for this anomaly. First, the antivirus tool may have incorrectly labeled one of the malware instances. Second, this particular malware may have the capabilities pertaining to both Groups 1 and 2, possibly due to a randomized choice of infection vector. We note that using additional malware detection products does not help to distinguish the two instances; all products assigned the same labels to these malware instances. A future work item is to dissect the malware binary to find out whether our second hypothesis is valid.

- Members of different malware families may correspond to the same alert pattern. For instance, Korgo.AF, Padobot.P, Rand, and Virut.AM have the same characterization using the IDS ruleset deployed in BotHunter during the time period for the experiment. A future work item we are investigating is extending the IDS rulebase to better differentiate them.

7 Summary and Future Work

This exploratory study indicates that the factor-analysis-based approach provides a promising means for behavioral characterization of bot infection instances. We were able to achieve such a characterization of 106 bot instances, harvested on a honeynet, which triggered 26 unique Snort signatures, as attributable to five factors. The factor loadings for the triggered signatures revealed seven patterns. These patterns, expressed in terms of a small number of high-level, mechanically derived quantities (or factors), can be used for malware characterization. Malware instances belonging to the same family generally have similiar factor patterns, and some malware instances correspond to distinguishing patterns (e.g., Sasser). We also identified a case in which rules can be safely removed from the IDS to improve performance with no loss in detection capability.

Based on the labels assigned by antivirus tools to the malware binaries studied in this paper, we observed that variants of a particular species of malware or malware families often exhibited similarities in the pattern of factor loadings. We also found some similarities between labeled malware and unlabeled malware, which could indicate an unknown common lineage or a new variant.

Because the analysis is achieved without using the source or destination address of the attack, it is suitable for use in a privacy-preserving repository such as the Cyber-TA repository.

We envision that a collaborative malware repository may collect and analyze the malware profiles contributed by various organizations using the techniques presented in this paper. Moreover, subscribers would be able to query the results in near real time. As such, they may be able to learn the characterization of a piece of malware according to its pattern of factor loadings, rather than waiting for a time- and analyst-intensive decompilation and labeling of new malware species. Potentially, this enables rapid deployment of defenses to counter the emerging malware.

Finally, we plan to refine our approach to address its weaknesses identified in Section 6, and investigate the usefulness of our approach for analyzing non-bot traffic and reports generated in various types of sensors.

Acknowledgments

This material is based on work supported through the U.S. Army Research Office (ARO) under the Cyber-TA Research Grant No. W911NF-06-1-0316. Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of U.S. ARO.

References

[1] BAILEY, M., OBERHEIDE, J., ANDERSEN, J., MAO, Z. M., JAHANIAN, F., AND NAZARIO, J. Automated classification and analysis of Internet malware. In *Recent Advances in Intrusion Detection (RAID 2007)* (Gold Coast, Australia, Sept. 5–7, 2007), C. Kruegel, R. Lippmann, and A. Clark, Eds., vol. 4637 of *LNCS*, Springer-Verlag, pp. 178–197.

[2] CHEUNG, S., LINDQVIST, U., AND FONG, M. W. Modeling multistep cyber attacks for scenario recognition. In *Proceedings of the* 3^{rd} *DARPA Information Survivability Conference and*

Exposition (DISCEX III) (Washington, D.C., Apr. 22–24, 2003), pp. 284–292.

[3] CUPPENS, F., AND ORTALO, R. LAMBDA: A language to model a database for detection of attacks. In *Recent Advances in Intrusion Detection (RAID 2000)* (Toulouse, France, Oct. 2–4, 2000), H. Debar, L. Mé, and S. F. Wu, Eds., vol. 1907 of *LNCS*, Springer-Verlag, pp. 197–216.

[4] GOLDMAN, R. P., HEIMERDINGER, W., HARP, S. A., GEIB, C. W., THOMAS, V., AND CARTER, R. L. Information modeling for intrusion report aggregation. In *DARPA Information Survivability Conference and Exposition (DISCEX II)* (Anaheim, California, June 12–14, 2001), vol. 1, pp. 329–342.

[5] GU, G., PORRAS, P., YEGNESWARAN, V., FONG, M., AND LEE, W. BotHunter: Detecting malware infection through IDS-driven dialog correlation. In *Proceedings of the 16th USENIX Security Symposium* (Boston, M.A., Aug. 2007), pp. 167–182.

[6] JULISCH, K. Mining alarm clusters to improve alarm handling efficiency. In *Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC 2001)* (New Orleans, Louisiana, Dec. 10–14, 2001), pp. 12–21.

[7] MICHEL, C., AND MÉ, L. ADeLe: An attack description language for knowledge-based intrusion detection. In *Trusted Information: The New Decade Challenge: IFIP TC11 16th International Conference on Information Security (IFIP/SEC'01)* (Paris, France, June 11–13, 2001), M. Dupuy and P. Paradinas, Eds., pp. 353–368.

[8] NING, P., JAJODIA, S., AND WANG, X. S. Abstraction-based intrusion detection in distributed environments. *ACM Transactions on Information and System Security 4*, 4 (Nov. 2001), 407–452.

[9] PORRAS, P. A., FONG, M. W., AND VALDES, A. A mission-impact-based approach to INFOSEC alarm correlation. In *Recent Advances in Intrusion Detection (RAID 2002)* (Zurich, Switzerland, Oct. 16–18, 2002), A. Wespi, G. Vigna, and L. Deri, Eds., vol. 2516 of *LNCS*, pp. 95–114.

[10] QIN, X., AND LEE, W. Statistical causality analysis of INFOSEC alert data. In *Recent Advances in Intrusion Detection (RAID 2003)* (Pittsburgh, PA, Sept. 8–10, 2003), G. Vigna, E. Jonsson, and C. Kruegel, Eds., vol. 2820 of *LNCS*.

[11] RENCHER, A. C. *Methods of Multivariate Analysis, 2nd Edition*. Wiley, 2002.

[12] STANIFORD-CHEN, S., CHEUNG, S., CRAWFORD, R., DILGER, M., FRANK, J., HOAGLAND, J., LEVITT, K., WEE, C., YIP, R., AND ZERKLE, D. GrIDS – A graph-based intrusion detection system for large networks. In *Proceedings of the 19th National Information Systems Security Conference* (1996).

[13] TAYLOR, A. A brief introduction to factor analysis (handout), Mar. 2004.

[14] TEMPLETON, S., AND LEVITT, K. A requires/provides model for computer attacks. In *Proceedings of the New Security Paradigms Workshop* (Cork, Ireland, 2000), pp. 31–40.

[15] TRUJILLO-ORTIZ, A., HERNANDEZ-WALLS, R., CASTRO-PEREZ, A., RODRIGUEZ-CEJA, M., MELENDEZ-SANCHEZ, A., DEL ANGEL-BUSTOS, E., MELO-ROSALES, M., VEGA-RODRIGUEZ, B., MORENO-MEDINA, C., RAMIREZ-VALDEZ, A., D'OLIVO-CORDERO, J., ESPINOSA-CHAURAND, L., AND BELTRAN-FLORES, G. Anfactpc: Factor analysis by the principal components method (a Matlab file), Mar. 2006.

[16] VALDES, A., AND SKINNER, K. Probabilistic alert correlation. In *Recent Advances in Intrusion Detection (RAID 2001)* (Davis, CA, Oct. 2001), W. Lee, L. Me, and A. Wespi, Eds., LNCS, pp. 54–68.

[17] VALEUR, F., VIGNA, G., KRUEGEL, C., AND KEMMERER, R. A comprehensive approach to intrusion detection alert correlation. *IEEE Transactions on Dependable and Secure Computing 1*, 3 (Jul.-Sept. 2004), 146–169.

[18] VIINIKKA, J., DEBAR, H., ME, L., AND SEGUIER, R. Time series modeling for IDS alert management. In *Proceedings of the ACM Symposium on Information, Computer and Communications Security (AsiaCCS '06)* (Taipei, Taiwan, Mar. 2006), pp. 102–113.

[19] YANG, J., NING, P., WANG, X. S., AND JAJODIA, S. CARDS: A distributed system for detecting coordinated attacks. In *Proceedings of IFIP TC11 16th Annual Working Conference on Information Security* (2000), pp. 171–180.

Notes

[1]We have performed factor analysis on a reduced dataset pertaining to 19 instead of 26 variables, keeping only one SID for each of the perfect correlation subsets. The factor analysis result for this reduced dataset is qualitatively the same as that of the original dataset.

[2]The latent root criterion, a commonly used technique for selecting the number of factors, chooses factors whose eigenvalues are greater than one. The intuition behind this technique is that these factors account for more variance than any individual variable in the original dataset does.

[3]When AntiVir returns "HEUR/Crypted" or "nothing", we try to use the label returned by Symantec.