Detection of Spam Hosts and Spam Bots Using Network Flow Traffic Modeling

Willa K. Ehrlich, Anestis Karasaridis, Danielle Liu, and David Hoeflin
AT&T Labs, Middletown, NJ, USA
e-mail: {wehrlich,karasaridis,dliu,dhoeflin} at att dot com

***Abstract*—In this paper, we present an approach for detecting e-mail spam originating hosts, spam bots and their respective controllers based on network flow data and DNS metadata. Our approach consists of first establishing SMTP traffic models of legitimate vs. spammer SMTP clients and then classifying unknown SMTP clients with respect to their current SMTP traffic distance from these models. An entropy-based traffic component extraction algorithm is then applied to traffic flows of hosts identified as e-mail spammers to determine whether their traffic profiles indicate that they are engaged in other exploits. Spam hosts that are determined to be compromised are processed further to determine their command-and-control using a two-stage approach that involves the calculation of several flow-based metrics, such as distance to common control traffic models, periodicity, and recurrent behavior. DNS passive replication metadata are analyzed to provide additional evidence of abnormal use of DNS to access suspected controllers. We illustrate our approach with examples of detected controllers in large HTTP(S) botnets such as `Cutwail`, `Ozdok` and `Zeus`, using flow data collected from our backbone network.**

I. Introduction

E-mail spam, also known as unsolicited bulk e-mail or unsolicited commercial e-mail, is the practice of sending unwanted e-mail messages frequently with commercial content in large quantities to an indiscriminate set of recipients. Spam is technically delivered the same way as legitimate e-mail, utilizing the Simple Mail Transfer Protocol (SMTP).

Currently, a large fraction of spam comes from botnets, i.e., large collections of compromised machines controlled by a single entity, with the implication that e-mail spam detection is an effective strategy for subsequent botnet detection. In this paper, we present an approach for identifying botnet command-and-control by first detecting e-mail spam originating hosts based on SMTP flow traffic characteristics. Spam hosts whose host traffic profiles indicate that they are compromised are processed further using real-time botnet analysis algorithms to identify centralized or distributed botnet controllers.

The paper is organized into the following sections: In Section II, we summarize related work on spam bot detection and describe the contribution of the current work. In Section III, we derive multivariate traffic models (based on network flow data) of spam and legitimate SMTP clients and present a Bayesian classification rule for classifying SMTP clients into spammers vs. legitimate e-mail clients. In Section IV, we analyze the modeling accuracy in classifying blacklisted and whitelisted SMTP clients. Section V describes our approach in automatically detecting controllers of the compromised spam hosts. Finally, Section VI provides a summary of the paper and conclusions.

II. Recent Work and Current Contribution

Two main approaches used currently to detect/mitigate spam include e-mail payload content filtering (e.g., [1, 14]), and address-based filtering [12]. In content filtering, the header and body of an e-mail are analyzed for certain keywords, patterns (e.g., URL strings), message signatures, and message authentication policies that are characteristic of e-mail spam. In address-based filtering, the originating IP address and session establishment data are analyzed for reputation, domain signature, connection authentication policy, session signature, protocol, traffic and connection limits. IP addresses of spam e-mail clients are entered into centrally maintained databases called Real-time Blackhole Lists (RBLs) or, if accessible via the Domain Name System (DNS), DNS blacklists (DNSBLs) such that Mail Transfer Agents (MTAs) can reject or throttle all mail either originating from or relayed by a listed host.

In the case of content filtering, blocking rules need to be updated frequently and new spam corpora must be used for re-training (if the keywords are learned dynamically by means of a Bayesian filter) as spammers devise new content and formats to circumvent the filters. A recent content-based approach [16] achieves low false positive rates for template-based spam generated by certain botnets, by deriving the very templates used to create the spam (in the form of regular expression signatures). However, it can be evaded by spam that uses multiple interleaving templates generated by different bots or spam that is not based on templates. In general, content analysis results in a higher degree of privacy intrusion and processing overhead. In the case of address-based filtering, if spammers use addresses without reputation (e.g., when the proportion of spam e-mail from dynamic addresses is significant [21], or if low-volume spamming occurs from spammers who are compromised hosts [17]) or if spam sources become more short-lived [19], then an address-based filtering approach based on blacklists will be less effective.

A social network based approach to spam detection applies a graph-theoretic analysis to interactions between e-mail addresses to construct a user's personal e-mail network [3, 4]. This approach requires header information of all the messages in a user's inbox; hence, it is considered invasive. A graph-theoretic approach for differentiating legitimate e-mail client MTAs that submit SMTP traffic to legitimate server MTAs

only vs. spam client MTAs that submit SMTP traffic both to legitimate server MTAs and to hosts that do not typically receive SMTP traffic, is presented in [5]. Since this work is based on SMTP transport header data, there is minimal privacy intrusion. However, the assumption that a spammer will also send SMTP traffic to "illegitimate" e-mail servers may not be warranted.

Several investigators have characterized spam vs. legitimate SMTP traffic and concluded that these two types of traffic are statistically different. For example, Gomes et al. [6], using e-mail server log data, indicated that the sizes of the legitimate e-mails are much more variable and have a heavier tail with spam messages exhibiting both lower average e-mail message size and less variation in e-mail message size. This characterization is supported by Schatzmann's et al. [18] flow analysis of SMTP traffic using netflow data collected from border routers of a major ISP and by Hao's et al. [10] analysis of McAfee's TrustedSource e-mail log data. However, the first two groups of investigators did not specify a procedure for classifying SMTP clients into e-mail spammers vs. legitimate SMTP clients so that their work cannot be directly applied to real-time SMTP client classification.

Some studies (e.g., [24, 22]) have approached the problem of spam botnet detection by identifying spam bots participating in the same spam e-mail campaign under the assumption that hosts participating in the same campaign are part of the same botnet. However, [11] demonstrated that this assumption is not always true since a single spam campaign is often carried by more one botnets.

There have been several general botnet detection algorithms (e.g., [7, 8, 9]) that identify suspect packets and host behaviors and take advantage of event correlation to report infected hosts and their likely controllers. Most of these studies rely on exhaustive deep packet inspection which can be expensive and have significant operational overhead in large networks.

A network flow based approach to detecting botnets, given a set of suspicious clients was proposed by Karasaridis et al. [13]. For a set of suspicious clients (e.g., hosts found scanning for vulnerabilities), flow records (in which the suspicious client's IP address is either the source or destination address) were obtained from multiple network links, analyzed and compared to IRC traffic models for suspected controller activity. This approach could be extended to uncover botnet controllers that use other control protocols to communicate with spam bots.

In this paper, we present an alternative approach to botnet detection based on characteristics of spam vs. legitimate SMTP traffic derivable from SMTP flow data and characteristics of compromised vs. non-compromised spammers obtained from other flow data. In our approach, we derive multivariate models of known spammer and known legitimate SMTP client traffic based on SMTP flows and then classify unknown SMTP clients based on their current SMTP traffic vectors' distance from these models. If a known spammer or an unknown SMTP client classified as a spammer by our spam classification algorithm is detected, the secondary behavior of spammers is profiled using a traffic extraction algorithm [23]. If the host traffic profiling indicates that the spammer is engaged in other exploits, the client is processed further to identify its botnet controller. Potential controllers are initially identified by applying several metrics (e.g., distance to common botnet control models, periodicity, recurrence) to flow traffic of hosts that interact with these compromised spammers. Subsequently, passive DNS replication metadata [20] are analyzed for additional evidence of abnormal use of DNS to obtain access to the suspected controllers.

III. Multivariate Models and Bayesian Classification of SMTP Clients based on Flow Data

A. Traffic Analysis of Spammers vs. Legitimate E-mail Clients

1) Flow Data Collection: Our analyzed data consist of SMTP flows that are aggregate traffic records between pairs of hosts. A flow record is a tuple that consists of the source and destination IP (*sip*/*dip*) addresses, the protocol (e.g., TCP, UDP), the source and destination ports (*sport*/*dport*), and other aggregated data such as the number of packets, and bytes transferred between the hosts and the TCP flags. In the case of an SMTP request, the source IP address corresponds to the SMTP client, the source port is an ephemeral port, the destination IP address corresponds to the SMTP server, the destination port is 25, and the protocol is TCP. In the case of an SMTP response the *sip* and *sport* are those of the SMTP server and the *dip* and *dport* are those of the SMTP client.

In the current context, our flow data refer to flows traversing links between our network and other ISPs (see Figure 1). Consequently, we define an SMTP client as the MTA in the sender AS that initiates an SMTP connection using a local ephemeral port. We define an SMTP server as the MTA in the receiving AS that accepts the SMTP connection on port 25/TCP to deliver the e-mail to its final destination.

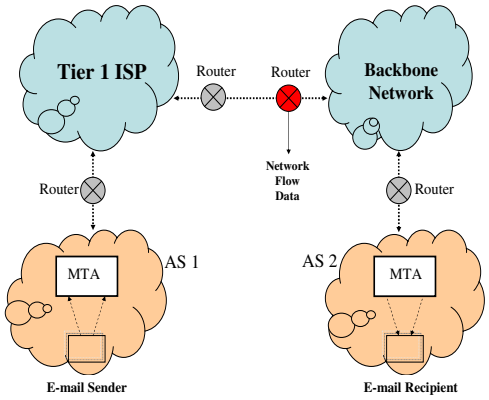

Fig. 1. High-level SMTP traffic paths and data collection points (in red).

In order to manage spam for e-mail services on our network, we maintain our own sets of blacklisted and whitelisted SMTP

clients. Our blacklisted SMTP clients are categorized into a daily updated list to capture the dynamically changing spammers and a less frequently updated list to capture the more static ones. Our whitelist contains "friendly" IP addresses that interact regularly with our mail gateway servers for legitimate purposes. Based on the lists that were in effect for a given calendar date/hour, we collected SMTP traffic flows traversing a diverse set of links for a set of known e-mail spammers and a set of known legitimate clients.

2) Important Discriminators of SMTP Traffic: To ensure that we are analyzing purposeful SMTP requests (as opposed to, for example, scans to destination TCP port 25 or incomplete 3-way handshakes), we consider flows that contain at a minimum the PUSH TCP flag.

The differentiation between spammers vs. legitimate SMTP clients is illustrated in Figure 2 based on flow data collected for 113 hours, where the average hourly number of flows analyzed was 947.2 Million (minimum=560.5M; maximum=1361.5M) over an average number of 27 peering links. Figure 2 indicates that for SMTP flows containing a PUSH flag, the distributions of e-mail message size (estimated by the number of bytes per flow (BPF)) originating from blacklisted vs. whitelisted clients are distinguishably different. Specifically, the payload byte sizes of SMTP request flows of the whitelisted SMTP clients are larger and much more variable than the sizes for blacklisted SMTP clients. Consequently, traffic models of SMTP traffic flows can be derived to distinguish the behaviors associated with spammers vs. legitimate SMTP clients. Summary statistics of the BPF for the two categories of SMTP clients presented in Figure 2 are given in Table I.

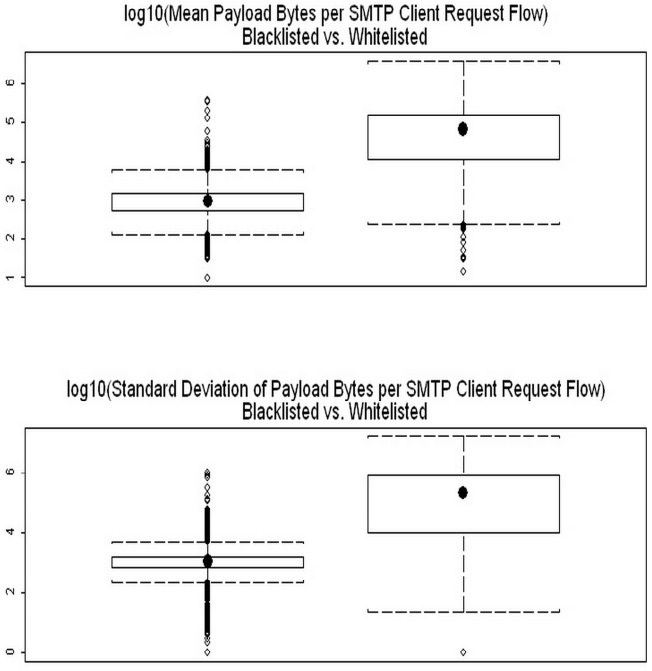

Fig. 2. Boxplot distributions of SMTP traffic for blacklisted and whitelisted clients.

TABLE I
DISTRIBUTION STATISTICS FOR WHITELISTED VS. BLACKLISTED SMTP CLIENTS. VARIABLE b DENOTES BYTES PER FLOW (BPF)

	Whitelisted SMTP clients		Blacklisted SMTP clients	
Number of hourly client sessions	3262		3388	
Statistic	$\log(\bar{b})$	$\log(\sigma(b))$	$\log(\bar{b})$	$\log(\sigma(b))$
Upper Extreme	6.60	7.20	3.78	3.72
Q3	5.20	5.89	3.16	3.20
Q2	4.82	5.31	2.96	3.03
Q1	4.06	4.00	2.74	2.85
Lower Extreme	2.36	1.38	2.11	2.33

B. Bayesian Classification

Consider a bivariate vector $\mathbf{x} = [x_1, x_2]$, associated with an SMTP client's observed traffic during a given time interval, where x_1 and x_2 are calculated from the logarithm of the mean and standard deviation of the client's BPF data, respectively. We wish to categorize this traffic vector into classes $c_j,\ j = 1, \ldots, J$ based on the expected traffic vector exhibited by e-mail spammers vs. legitimate SMTP clients. A Bayesian statistical decision $C(\mathbf{x}) = c_j$ about the class of a data point $\mathbf{x}$ is based on $P(c_j/\mathbf{x})$, the probability of class c_j conditional on the observation $\mathbf{x}$. This probability depends on $P(c_j)$, the probability of class c_j independently of the observed data (the prior probability), and $P(\mathbf{x}/c_j)$, the conditional distribution function of $\mathbf{x}$, given that is coming from class c_j.

In the current context, where $J = 2$, a decision can be made with respect to classifying an SMTP client as e-mail spammer, whenever

$$\frac{P(c_S) * P(\mathbf{x}/c_S)}{P(c_S) * P(\mathbf{x}/c_S) + P(c_L) * P(\mathbf{x}/c_L)} > T, \qquad (1)$$

where c_S and c_L denote the spammer and legitimate classes (i.e., $c_1 = c_L$ and $c_2 = c_S$), respectively and T is a threshold. By varying T, one can allow for less false positives (incorrectly classifying legitimate clients as spammers) at the expense of fewer true positives (i.e., correctly classifying spammers) or vice versa. Since we do not have bias for either class, we assign equal prior probabilities to the two classes (i.e., $P(c_S) = P(c_L)$), and so we can write condition (1) of spammer classification as:

$$P(\mathbf{x}/c_S)/((P\mathbf{x}/c_S) + P(\mathbf{x}/c_L)) > T. \qquad (2)$$

The probabilities $P(\mathbf{x}/c_j)$ are calculated using the bivariate normal distribution function modeled from SMTP clients of the respective class.

IV. MULTIVARIATE TRAFFIC MODEL VALIDATION

A. Stability of Traffic Model Parameters Over Time

For a given class of SMTP client there are five parameters that define the SMTP traffic model. These parameters are defined in Table II. For both classes of SMTP clients, the two traffic variables are positively correlated, with a correlation

TABLE II
TRAFFIC MODEL PARAMETERS FOR A GIVEN SMTP CLIENT CLASS

Parameter Notation	Parameter Interpretation
$\mu_{X_{1j}}$, $j = 1, 2$	$\mu_{X_{1j}} = E_j[\log Y_{1ij}]$, Y_{1ij}: mean (across flows) of BPF for client i in class j
$\mu_{X_{2j}}$, $j = 1, 2$	$\mu_{X_{2j}} = E_j[\log Y_{2ij}]$, Y_{2ij}: stddev (across flows) of BPF for client i in class j
$\text{Var}(X_{1j})$	$\sigma^2_{X_{1j}} = E_j(\log Y_{1ij} - \mu_{X_{1j}})^2$
$\text{Var}(X_{2j})$	$\sigma^2_{X_{2j}} = E_j(\log Y_{2ij} - \mu_{X_{2j}})^2$
$\text{Cov}(X_{1j}X_{2j})$	$E_j(\log Y_{1ij} - \mu_{X_{1j}})(\log Y_{2ij} - \mu_{X_{2j}})$

coefficient of 0.95 for whitelisted and 0.74 for blacklisted SMTP clients. This implies that a multivariate normal model that explicitly addresses dependency between variables in terms of a covariance matrix, is well suited for the current application.

Time series analysis of these parameter values by SMTP client type[1] indicated a periodicity effect in SMTP traffic generated by legitimate SMTP clients. A scatter plot of traffic model parameter values as a function of hour of day and day of week for the two classes of SMTP clients indicated that for legitimate SMTP clients ($j = 1$), both the expected (across clients) average SMTP request flow payload bytes size ($\mu_{X_{11}}$) and the expected standard deviation of the SMTP request flow payload bytes size ($\mu_{X_{21}}$) are greatest at 16:00 UTC time with the exception of Sunday. In contrast, both the variances and covariance (i.e., $\text{Var}(X_{11}), \text{Var}(X_{21}), \text{Cov}(X_{11}X_{21})$) of these two SMTP message size characteristics are lowest at 16:00 UTC time, again with the exception of Sunday. These types of patterns are much less pronounced for the blacklisted SMTP clients ($j = 2$). This pattern is consistent with [6] in that traditional e-mail arrivals exhibit a daily cycle and thus have high rates during certain times of the day, in contrast to the more homogeneous arrival rates of spam e-mails.

B. Adjusting Traffic Model Parameters Using Exponentially Weighted Moving Average (EWMA) Smoothing

Given the existence of a periodicity effect associated with time of day and day of week, we characterize a seasonality cycle of one week duration corresponding to 21 successive 8-hour time periods. We define a set of traffic model parameter values for a given SMTP client type for each of these 21 time periods (corresponding to a period of one week) and apply exponentially weighted moving average (EWMA) to smooth short-term fluctuations associated with model parameter values[2]. Since we did not observe any sudden fluctuations in the parameter values, we set the smoothing parameter (i.e., the weight of the current parameter value) α of EWMA to 0.5 (past parameter estimate is weighted by 1-α=0.5).

[1]Each time series contained, for a given SMTP Client type, values computed for a given traffic model parameter for 300 consecutive time periods, where each time period corresponded to a UTC time of 0:00; 08:00 or 16:00 for a given day of the week.

[2]An autoregressive model was also applied to these time series of model parameter values and then compared with the EWMA smoothing approach. Since the EWMA smoothing was comparable in terms of the mean squared error and is simpler to implement, we only present the results from the EWMA smoothing.

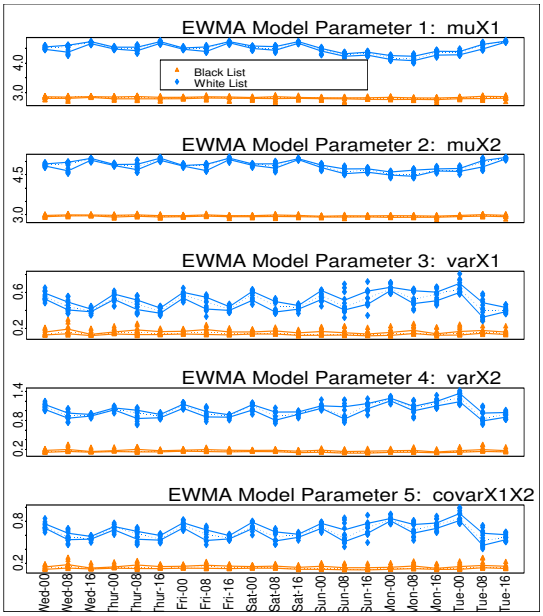

Fig. 3. Scatter plots of traffic model parameter values.

Figure 3 presents scatter plots of the traffic model parameter values as a function of time for a period of one week based on the EWMA filtering. Dashed lines indicate median parameter values while solid lines indicate the 25th and 75th quartile parameter values. For a given time of day and day of week, for model parameters $\mu_{X_{1j}}, \mu_{X_{2j}}, \sigma^2_{X_{1j}}$, and $\sigma^2_{X_{2j}}$, the effect of the EWMA filtering is to reduce the variation in model parameter values (i.e., reduce the model parameter inter-quartile range) so that the two populations of SMTP clients are more distinguishable. Consequently, we utilize the EWMA parameter values when evaluating the accuracy of the traffic models in classifying SMTP clients.

C. Accuracy of Traffic Models in Classifying Blacklisted vs. Whitelisted SMTP Clients

We applied the following four metrics to evaluate model classification accuracy:

- P(Classify Spammer/Blacklisted SMTP client): the ratio of correctly classified spammers to all blacklisted SMTP clients.
- P(Classify Legitimate/Blacklisted SMTP client): the ratio of blacklisted SMTP clients incorrectly classified as legitimate to all blacklisted SMTP clients.
- P(Classify Legitimate/Whitelisted SMTP client): the ratio of correctly classified legitimate SMTP clients to all whitelisted SMTP clients.

TABLE III
EVALUATION OF SMTP TRAFFIC MODEL CLASSIFICATION ACCURACY

T	P(Classify Spammer/ Blacklisted)	P(Classify Legitimate/ Blacklisted)	P(Classify Legitimate/ Whitelisted)	P(Classify Spammer/ Whitelisted)
0.8	0.887	0.031	0.864	0.05
0.85	0.862	0.027	0.852	0.042
0.9	0.817	0.023	0.836	0.032
0.95	0.708	0.018	0.808	0.018

- P(Classify Spammer/Whitelisted SMTP client): the ratio of whitelisted SMTP clients incorrectly classified as spammers to all whitelisted SMTP clients

The median values for each of these four metrics are given in Table III for different threshold values, T. Table III demonstrates that we can reduce the false positives by increasing the threshold value T, though at the expense of reduced true positives.

V. DETECTION OF SPAM BOT CONTROLLERS

The proliferation of botnets is driven to a large extent by their capability to automate large spam campaigns. Since a portion of a botnet is expected to be used for spamming, we can use our spam host detection algorithm to uncover possible spam bots and the botnets that they belong to. In this section, we demonstrate the application of our spammer classification algorithm in identifying spam bots and their controllers.

A. Identifying Compromised E-mail Spammers by Host Traffic Profiling

1) Host Traffic Profiling (HTP) Description: We applied an entropy-based significant traffic component extraction procedure to flows collected for detected spammers [23]. When extracting a set of significant local and remote ports for any given spamming host, we assume that the probability distribution of the target variables obeys a power law so that only relatively few values have significantly larger probabilities while the remaining values are close to being uniformly distributed. The procedure is first applied to extract the set of significant local ports and then the set of significant remote ports. As a metric of significance of a discrete random variable X, we use its normalized entropy which is defined as

$$H_n(X) = -\frac{\sum_i p(x_i)\log(p(x_i))}{\log(\min(N_x, m))}, \quad (3)$$

where $p(x_i)$ denotes the probability of discrete random variable x_i, m is the sample size and N_x is the number of all possible values of the discrete random variable. To interpret the significant traffic components of a spamming host, we analyze the set of flows that share the same port and compute H_n for each of the two remaining free dimensions (i.e., remote hosts and remote ports, or remote hosts and local ports).

An example of a host traffic profile computed for a whitelisted SMTP client is given in Table IV[3]. The profile indicates that the host initiates SMTP interactions with remote hosts (i.e., on remote TCP port 25) and that it also initiates DNS interactions with remote hosts (i.e., on remote UDP port 53) using local UDP port 53.

[3]For brevity in this example, we omitted from the table remote hosts (since none were significant) and traffic statistics.

TABLE IV
EXAMPLE HOST TRAFFIC PROFILE FOR A WHITELISTED SMTP CLIENT

Traffic Component	Entropy of Local Ports	Significant Local Ports	Entropy of Remote Hosts	Interpretation
Remote Port 25/tcp	0.972	N/A	0.622	e-mail client
Remote Port 53/udp	0	UDP-53	0.788	Host Interacting via local port 53/udp with remote port 53/udp

2) Host Traffic Profiles of Categories of E-mail Clients: Analysis of the host traffic profiles constructed for a set of known whitelisted clients and a set of known blacklisted clients observed for 21 hourly time periods indicated that, in addition to mail-related (tcp/25, tcp/110) services, these well-known e-mail clients exhibited or utilized DNS-related services (access to udp/53 and tcp/53), and issued or received ICMP traffic with message types other than "port unreachable". Consequently, we consider DNS-related services and non-port unreachable ICMP traffic, as possible traffic that a non-compromised SMTP client might be receiving or sending (i.e., "mail-related"). In contrast, all other services and port unreachable ICMP traffic are deemed "non-mail-related". These non-mail host traffic components include scanning activities (for malware propagation), binary downloading (for malware installs/updates), DoS attacks, other exploits, and command and control operations.

Table V presents the number of (known) blacklisted vs. (known) whitelisted e-mail clients with host traffic profiles containing non-mail-related traffic vs. the number of e-mail clients whose host traffic profiles contained mail-related traffic only. To analyze the dependency of non-mail-related traffic on the type of SMTP client, we performed an odds ratio test [2]. In the current context, the odds ratio represents the odds of non-mail-related traffic profile (signifying a likely compromised machine) occurring for one category of SMTP clients vs. the other. An odds ratio of 1 implies that the "possibly compromised" traffic behavior is independent of the SMTP client type.

The odds ratio for blacklisted vs. whitelisted e-mail clients in Table V is 5.46 confirming that whitelisted SMTP clients represent well-known dedicated e-mail hosts that are less likely to be involved in non-mail-related activities. Consequently, we can utilize host traffic profiling to classify detected e-mail spam hosts as compromised and use them as seeds for possible botnet activity.

TABLE V
NUMBER OF E-MAIL CLIENT SESSIONS BY CLIENT TYPE AND HOST TRAFFIC PROFILE (HTP) TYPE

SMTP Client Class	Hosts having HTP with non-mail components	Hosts having HTP with mail components
Blacklisted	1108	2157
Whitelisted	262	2786

B. Uncovering Botnet Controllers from Compromised Spam Hosts

Given that malware infected spam hosts typically show a network behavior markedly different from a regular SMTP gateway, we exploit this property to investigate if they are part of a botnet and identify their controllers. We do this by analyzing the flow records of the suspicious spam hosts and the DNS metadata of the suspected controllers using a DNS passive replication database. A DNS analysis of the suspected controllers provides additional insight and confidence for potentially malicious activity. This can be done on-the-wire in near real-time without the need to collect and analyze large bodies of spam messages or the content of the communication with every suspected controller.

We direct our attention here to the most common botnet control mechanisms where there are distinctive controllers, which can be centralized or distributed, using standard application protocols such as HTTP, HTTPS and IRC. Note that even though these protocols are used to build general model frameworks of flow traffic, they are used in combination with other metrics that can unveil customized control mechanisms by giving higher tolerance to the distance metric, as we will discuss. To identify potential controllers, we applied a two-stage flow-based approach [13], which is outlined as follows:

Stage A:

1) For a given time period (e.g., 1 hour), obtain a set of blacklisted SMTP clients (from a daily updated upstream database) and a set of SMTP clients classified by the spam detection algorithm as spam hosts.
2) For each host identified above, obtain flow records from multiple network links where the IP address is either the source or destination address in the flow record.
3) Apply host traffic profiling to these flow records to identify malware-infected spam hosts
4) Process flow records associated with malware-infected spam hosts to identify flows representing communication to a possible controller and summarize these interactions as candidate controller conversations containing client (i.e., infected spam host) and server (i.e., controller) IPs, server port, number of flows, packets, bytes exchanged and the start and end time of the conversation.

Stage B:

1) Aggregate candidate controller conversations (optionally for longer periods than the flow summarization period, e.g., 1 day), rank server addresses/ports by the number of suspicious clients and calculate distance of these candidate controller conversations to the traffic model. The models consist of quartiles of flows per client, number of packets and number of bytes and are defined for typical IRC, HTTP and HTTPS control traffic[4] for both directions of a connection (requests and responses). For a suspect server/port pair that satisfies a minimum threshold of clients and has small distance relative to a model we analyze the flows in more detail and calculate the following additional components: *i)* number of quasi-periodic clients, and *ii)* number of zero-entropy clients. Quasi-periodic clients are clients whose flow records have approximately periodic interarrival times. Zero-entropy clients are clients whose flows with suspected controllers have repetitive patterns of packet and byte counts and flags for a given protocol.
2) Perform DNS metadata analysis for suspected controllers. The DNS metadata contain known mappings between a fully-qualified domain name, the resolved IP address, a count of responses with the same resolution, and the start and end time of all known resolutions of the same pair domain–address. The DNS analysis of the IP address of the suspected controller provides the following output: *i)* count of all domains resolved to the address historically, *ii)* count of domains that resolved to the address recently (e.g., last 1 day), and *iii)* number of transient domains related to the suspect address. Transient domains are domains that migrate frequently between diverse provider addresses, indicating an evasion effort. To determine the transiency of a domain we consider the average time overlap between addresses for the same domain and the diversity of the addresses in terms of AS numbers and IP registration data.
3) Assign a botnet controller confidence score to each suspected controller–port pair based on factors (with increasing weight) such as the number of suspicious spam host clients connected, number of quasi-periodic clients, number of zero-entropy clients, and number of transient domains. The score is also increased if there were no domains ever mapped to the suspected controller (i.e., accessed only by its address). The overall score is a linear weighted function of these factors. The weights are adjusted for different average volumes of suspicious clients, flow records and time interval of summary record aggregation. Alarm records are generated whenever the confidence score exceeds a threshold (e.g., 100).

In the following, we give some examples of controllers detected using the approach described above and explain the different metrics that contributed to the confidence score.

1) Discussion of results for controller detection: Figure 4 gives an example of a suspected controller record. This particular record identifies a host[5] that is a controller of the `Ozdok` botnet that is mainly responsible for conducting

[4]Models can also be derived for known botnets such as Waledac.
[5]All IP addresses in this paper have been anonymized.

```
#Server_IP|Server_Port|Number_of_Suspicious_Clients|
 Port_Euclidean_Distance|Number_of_Suspicious_Periodic_
 Clients|0-EntropyClients|Number_of_domains|Number_of_
 recent_domains|Number_of_Transient_domains|Aggregate_Score
10.232.229.114|80|3|78.80|1|3|4|1|0|115
Date/Hr Timestamp: 2010012504
Recent domain used=selementusaks.org
Total number of domains that have mapped to this IP=4
Number of domains that have recently mapped to this IP=1
Examining domain selementusaks.org
Number of historical IPs=0
Domain has a small number of historical IPs (0 <2)
```

Fig. 4. Alarm record for a suspected controller of the Ozdok spam botnet. One of the compromised spam hosts shows a quasi-periodic behavior while three of the spamming hosts show a recurrent behavior towards the controller.

```
#Server_IP|Server_Port|Number_of_Suspicious_Clients|
 Port_Euclidean_Distance|Number_of_Suspicious_Periodic_
 Clients|0-EntropyClients|Number_of_domains|Number_of_
 recent_domains|Number_of_Transient_domains|Aggregate_Score
10.19.191.55|443|2|36.41|0|1|0|0|0|85
Date/Hr Timestamp: 2009120701
Total number of domains that have mapped to this IP=0
Number of domains that have recently mapped to this IP=0
There are no domains associated currently with this
 IP address
```

Fig. 5. Alarm record for a suspected controller that is part of the Cutwail spam botnet. The controller is accessed directly by its IP address since there were no records of domains resolved to its address.

large spam campaigns. The record indicates the IP address (anonymized) and the port of a suspect controller. Metrics of interest in this case are the number of quasi-periodic clients and the number of zero-entropy clients. Notice that the record was based only on 3 suspicious clients (compromised spammers). Of these, 1 exhibited periodic patterns and all 3 of them were zero-entropy, i.e., they had repetitive bytes, packets and flags in their flow records. We found that the suspect IP address was resolved historically to by 4 different domains but only 1 of them has been used recently. The currently used domain is not transient and it has no other historical addresses other than the currently used one.

Figure 5 illustrates an example where the suspected controller operates on port 443/tcp and is one of the distributed controllers for the botnet Cutwail. In this case, the commands are communicated via HTTPS. There are two interesting characteristics in the report: *a)* one of the two suspicious clients has zero-entropy, and *b)* the suspected controller IP address was never associated to any domain name.

In the example of Figure 6, the distinctive feature is that there are 2 transient domains associated with the suspected botnet controller. Further investigation indicated that the suspected controller was commanding machines controlled by the Zeus botnet responsible for leaking internal corporate data. As we see, there was no overlap in time between the different addresses used for one of the transient domains. Also, the average IP distance had the maximum possible value of 1, which means that all addresses had different /24 prefixes, address registration data and AS numbers.

2) Discussion of false positives in controller detection and possible evasion techniques: Using flow and DNS metadata we are able to detect automatically IRC controllers in near

```
10.51.196.242|80|2|45.36|0|0|6064|63|2|110
Date/Hr Timestamp: 2010020304
Recent domain used=lambert.66ghz.com
Recent domain used=zal.te.ua,
...
Examining domain lambert.66ghz.com
Number of historical IPs=0
Domain has a small number of  historical IPs (0 <2)
...
Examining domain zal.te.ua
Number of historical IPs=2
Average Time Overlap between IPs=0.000
Average IP distance=1.000
Domain zal.te.ua appears to be transient
#Domain|IPaddress|NumResponses|NumDomainstoIP|
 StartTime|EndTime|Lifespan(Days)
10.51.196.242|4|6064|20091112@23:25:21|20100129@23:43:01|78
10.43.65.6|5|307|20090113@15:24:39|20090510@15:41:05|117
10.32.73.138|4|233|20081027@07:41:07|20081127@15:53:43|31.3
```

Fig. 6. Alarm record for a controller of the Zeus botnet. The address of the controller is linked to two domains that appear to be transient.

real-time with virtually no false positives. DNS metadata analysis improves the detection precision from about 98% [13] to 100%. Normal control IRC traffic can be accurately modeled using flow data, which makes the distance calculation a good predictor of control in combination with the other metrics we use.

Botnet detection where controllers use HTTP(S) presents a bigger challenge if the analysis is based solely on flow records. The main reasons are: *a)* the communication between the bot and its controllers can be highly variable due to being coded in many different ways by the different malware authors, *b)* bots often communicate with legitimate sites for a variety of reasons, e.g., to check for the latest news (to use for example in spam email), to get accurate time and timezone, to check for host connectivity, etc. Many times this behavior has similar characteristics as control traffic, *c)* HTTP(S) traffic is much more common and therefore control traffic can be hidden with more ease in legitimate traffic, and *d)* control typically is more distributed and therefore we detect smaller sets of suspicious clients connecting to questionable servers.

However, we found that supplemental analysis of the DNS metadata adds significant confidence to the detection. For example, connections to suspected controller IPs that do not have any history of domain names pointing to them or that point to consumer dynamic pool domains, short lived or transient domains, lead most of the time to malware hub servers validated by our internal analysis or external reports. Our approach has detected automatically distributed controllers for some of the largest spam HTTP/HTTPS botnets such as Cutwail, Ozdok, Zeus and Waledac.

Security threats and countermeasures are in an accelerating arms race. New technologies are developed to detect today's threats but many new threats are able to evade many AV software packages and IDSs. The detection techniques we discussed have been proven effective in detecting C&C points of fast evolving malware propagating via a large scale network. However, we can envision an increased sophistication of the malware authors that can possibly challenge our assumptions and circumvent some of our filters. For example, spam host de-

tection can be challenged by spammers using legitimate SMTP servers (e.g., via Web mail accounts) to launder reputation and traffic statistics. This requires a significant amount of effort from spammers in setting up and authenticating to accounts on legitimate mail services which does not scale well and can be easily shut down by watchful system administrators. Alternatively, spammers can form spam messages in a way that they match the size statistics of normal email, i.e., have higher coefficient of variation and higher mean sizes. However, if the perpetrating hosts are compromised and part of a botnet, then it would be difficult to hide other traffic components that do not match the behavior of a normal SMTP gateway, i.e., they would need open active ports, locally or remotely, to receive commands or software updates, or they would exhibit background traffic related to activities of the legitimate users. The availability of flow records other than those corresponding to SMTP traffic, allow us to identify such difficult to hide traffic components from hosts that are involved in malicious activities.

Malware authors can also change the protocols and the way they use them to evade detection of controllers. Botnet control using for example HTTP already presents challenges due to the volume of normal HTTP traffic and the variability in how the protocol is used to communicate commands between the bots and the server. This necessitates the use of new traffic models. Also the distance to a model can only be used as a guideline and should not be a significant contributor to an overall confidence score. As we saw in the examples we gave, other metrics, different than the model distance, become more critical (e.g., DNS analysis) in identifying controllers with higher confidence.

VI. Conclusion

In this paper, we have presented a comprehensive approach for detecting spam bots and their respective controllers using flow data and DNS metadata. Our approach consists of establishing SMTP traffic models of legitimate vs. spammer SMTP clients and then classifying unknown SMTP clients using a Bayesian approach. We demonstrate that the likelihood of observing non-mail-related traffic behavior (indicative of compromised activities) among known spammers is much higher than for known legitimate clients.

Furthermore, we show how to further analyze the flow data of compromised spammers to find if they are part of a botnet and identify their likely C&C servers. This approach can be performed in near real-time and is automated. It is scalable since it does not depend on large number of compromised spammers to find likely controllers and does not depend on collecting and training on large sets of email spam.

Acknowledgments

The authors would like to give special thanks (alphabetically) to David Gross, John Hogoboom, Brian Rexroad, Chaim Spielman, and Stephen Wood for their input in this work and their comments on the paper.

References

[1] J. Blosser and D. Josephsen, "Scalable Centralized Bayesian Spam Mitigation with Bogofilter," Proc. of LISA '04: Eighteenth Systems Administration Conference, 2004, pps. 1-20.

[2] Y. M. M. Bishop, S. E. Fienberg and P. W. Holland, Discrete Multivariate Analysis: Theory and Practice, MIT Press, Cambridge, MA, 1976.

[3] P.O. Boykin and V. P. Roychowdhury, "Leveraging Social Networks to Fight Spam," Computer, 38 (4), 2005, pps. 61-68.

[4] P. A. Chirita, J. Diederich and W. Nejdl, "MailRank: Using Ranking for Spam Detection," Proc. of the 14th Intl. CIKM Conference on Information and Knowledge Management, CIKM2005, 2005.

[5] P. Desikan, J. Srivastava, "Analyzing Network Traffic to Detect E-mail Spamming Machines," Proc. ICDM Workshop on Privacy and Security Aspects of Data Mining, 2004, pps. 67-76.

[6] L.H. Gomes, C. Cazita, J. M. Almeida, V. Almeida and W. Meira Jr., "Characterizing a Spam Traffic," Proc. ACM SIGCOMM. Internet Measurement Conference, 2004.

[7] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation", Proc. 16th USENIX Security Symposium, 2007.

[8] G. Gu, J. Zhang, and Wenke Lee, "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic", Proc. of the 15th Annual Network and Distributed System Security Symposium (NDSS'08), 2008.

[9] G. Gu, R. Perdisci, J. Zhang, and Wenke Lee, "BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection", Proc. 17th USENIX Security Symposium, 2008.

[10] S. Hao, N. A. Syed, N. Feamster, A. G. Gray and S. Krasses, "Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine," 18th USENIX Security Symposium, 2009, pps 101-117.

[11] J. P. John, A. Moshchuk, S. D. Gribble and A. Krishnamurthy, "Studying Spamming Botnets Using Botlab," 6th USENIX Symposium on Network Systems Design and Implementation, 2009, pps. 291-306.

[12] J. Jung and E. Sit, "An Empirical Study of Spam Traffic and the Use of DNS blacklists," Proc. of the 4th ACM SIGCOMM Conference on Internet Measurement, 2004, pps. 370-375.

[13] A. Karasaridis, B. Rexroad, and D. Hoeflin, "Wide-scale Botnet Detection and Characterization," USENIX First Workshop on Hot Topics in Understanding Botnets (HotBots '07), 2007.

[14] J. Kim, K. Chung and K. Choi, "Spam Filtering With Dynamically Updated URL Statistics," IEEE Security & Privacy, 5(4), 2007, pps. 33-39.

[15] A. W. Moore and D. Zuev, "Internet Traffic Classification Using Bayesian Analysis Techniques," Proc. of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, 2005.

[16] A. Pitsillidis, K. Levchenko, C. Kreibich, C. Kanich, G. M. Voelker, V. Paxson, N. Weaver and S. Savage, "Botnet Judo: Fighting Spam with Itself," Proc. of the 17th Annual Network and Distributed System Security Symposium (NDSS), 2010.

[17] A. Ramachandran and N. Feamster, "Understanding the Network-Level Behavior of Spammers, SIGCOMM'06, 2006.

[18] D. Schatzmann, M. Burkhart and T. Spyropoulos, "Flow-level Characteristics of Spam and Ham," TIK-Report No. 291, Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland, 2008.

[19] S. Venkataraman , S. Sen , O. Spatscheck , P. Haffner and D. Song, "Exploiting Network Structure for Proactive Spam Mitigation," Proc. of 16th USENIX Security Symposium, 2007, pps. 1-18.

[20] F. Weimer, "Passive DNS Replication", 17th FIRST Annual Conference, Singapore, 2005.

[21] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber, "How Dynamic are IP Addresses? " SIGCOMM'07, 2007.

[22] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten and I. Osipkov, "Spamming Botnets: Signatures and Characteristics," SIGCOMM'08, 2008.

[23] K. Xu, Z-L Zhang and S. Bhattacharyya, "Profiling Internet Backbone Traffic: Behavior Models and Applications, "SIGCOMM'05, 2005, pps. 169 - 180.

[24] L. Zhuang, J. Dunagan, D. R. Simon, H. J. Wang, I. Osipkov, G. Hulten and J. D. Tygar, "Characterizing Botnets from e-mail Spam Records," LEET'08: First USENIX Workshop on Large-Scale Exploits and Emergent Threats, 2008.