An Analysis of Rogue AV Campaigns ⋆

Marco Cova[1], Corrado Leita[2], Olivier Thonnard[3],
Angelos D. Keromytis[4], and Marc Dacier[2]

[1] University of California Santa Barbara, Santa Barbara, USA, marco@cs.ucsb.edu
[2] Symantec Research Labs, Sophia Antipolis, France,
{corrado_leita, marc_dacier}@symantec.com
[3] Royal Military Academy, Brussels, Belgium, olivier.thonnard@rma.ac.be
[4] Columbia University, New York, USA, angelos@cs.columbia.edu

Abstract. Rogue antivirus software has recently received extensive attention, justified by the diffusion and efficacy of its propagation. We present a longitudinal analysis of the rogue antivirus threat ecosystem, focusing on the structure and dynamics of this threat and its economics. To that end, we compiled and mined a large dataset of characteristics of rogue antivirus domains and of the servers that host them.
The contributions of this paper are threefold. Firstly, we offer the first, to our knowledge, broad analysis of the infrastructure underpinning the distribution of rogue security software by tracking 6,500 malicious domains. Secondly, we show how to apply attack attribution methodologies to correlate campaigns likely to be associated to the same individuals or groups. By using these techniques, we identify 127 rogue security software campaigns comprising 4,549 domains. Finally, we contextualize our findings by comparing them to a different threat ecosystem, that of browser exploits. We underline the profound difference in the structure of the two threats, and we investigate the root causes of this difference by analyzing the economic balance of the rogue antivirus ecosystem. We track 372,096 victims over a period of 2 months and we take advantage of this information to retrieve monetization insights. While applied to a specific threat type, the methodology and the lessons learned from this work are of general applicability to develop a better understanding of the threat economies.

1 Introduction

A rogue security software program is a type of misleading application that pretends to be legitimate security software, such as an anti-virus scanner, but which

⋆ This work has been partially supported by the European Commission through project FP7-ICT-216026-WOMBAT funded by the 7th framework program. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission. This work was also partly supported by ONR through Grant N00014-07-1-0907 and the NSF through Grant CNS-09-14845. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ONR or the NSF. The work of Marco Cova was supported by a fellowship made possible by Symantec Research Labs.

actually provides the user with little or no protection. In some cases, rogue security software (heretofore referred to as "rogue AV") actually facilitates the installation of the very malicious code that it purports to protect against.

Rogue AVs typically find their way into victim machines in two ways. First, social engineering techniques can be used to convince inexperienced users that a rogue tool is legitimate and that its use is necessary to remediate often nonexistent or exaggerated threats found on the victim's computer. A second, stealthier technique consists of attracting victims to malicious web sites that exploit vulnerabilities in the client software (typically, the browser or one of its plugins) to download and install the rogue programs without any user intervention (*e.g.,* through *drive-by* downloads). After a rogue AV is installed on a victim's machine, it uses a number of techniques to convince (or force) a user to pay for additional tools or services, such as a "full version" of the program or the subscription to an update service. The cost of these additional programs or services ranges from $30–$100 [8].

In the last few years, rogue AVs have become a major security threat, both in terms of their pervasiveness and their financial impact. For example, over a 1-year period, Symantec's sensors detected 43 million installation attempts, covering over 250 distinct families of rogue AV software [8]. In addition, an investigation by Krebs revealed that affiliate programs alone can generate upward of $300,000 a month for the individuals that distribute rogue AVs [14].

As a consequence, different companies in the computer security industry have recently focused their attention on this threat [1,4,8]. Most of the existing works have considered individual facets of the rogue AV problem, for example, the malware code (*e.g.,* the installation techniques it employs), the sites involved in its distribution (*e.g.,* their number and geolocation), and the victims that it affects. However, little has been done to understand the rogue AV phenomenon as a whole, that is, relating how these individual pieces become combined in rogue AV campaigns.

We seek to fill this gap by providing a better understanding of the organization and dynamics of rogue AV campaigns. In particular, we focus on characterizing the infrastructure used in a campaign (*e.g.,* its web servers, DNS servers, and web sites) and the strategies used to create and manage it. We also investigate the uniqueness of our findings to this very specific threat type, and we investigate the motivations underneath these differences by exploring its economics.

The key of our approach is a method that, given a list of individual AV-hosting sites, allows us to group them into campaigns, each characterized by coherent features. More precisely, we use an extensive dataset including network and domain registration information, as well as network-observable temporal characteristics of a large number of domains that are associated with rogue AV advertising and distribution. To that dataset we apply a multi-criteria fusion algorithm to group together nodes based on a certain number of common elements likely due to the same root cause (*e.g.,* the same rogue AV campaign). This attribution method uses a combination of unsupervised, graph-based clustering combined with a data aggregation method inspired by multi-criteria decision

analysis (MCDA). On the one hand, our approach enables the identification of rogue AV campaigns and the analysis of the *modus operandi* of the individuals controlling them. On the other hand, this approach enables the execution of comparative analyses and the assessment of the uniqueness of the findings to the specific threat landscape.

The specific contributions of our work described in this paper include:

- The first, to our knowledge, large-scale analysis of the rogue AV threat and of its distribution infrastructure, conducted by tracking 6,500 malicious domains.
- A demonstration of the usefulness of attack attribution approaches to the problem of mining large security datasets. We show how, by using MCDA, we are able to discover specific campaigns likely to be associated to the action of a specific individual or group.
- The first characterization of the behavior of rogue AV campaigns and their economics. We reveal insights on the automated deployment of large amounts of domains, and we demonstrate their specificity to the threat landscape by comparing the results to those associated with other web-born threats. We collect information on 372,096 users (clients) interacting with some rogue AV domains to generate information on the average conversion rate of a rogue AV campaign. We demonstrate the existence of a very specific economic balance, that justifies a bigger investment in the deployment and maintenance of such large-scale campaigns.

The remainder of this paper is organized as follows. Section 2 describes the state of the art on tracking and mitigating the rogue AV threat. Section 3 describes the features we used in our analysis, as well as the clustering technique itself. Section 4 highlights our most interesting insights following the analysis, while Section 5 assesses the specificity of our findings to the rogue AV threat, and looks into their economic motivations. Finally, Section 6 summarizes some of the lessons we learned from this study, and Section 7 concludes the document.

2 State of the Art

The presence of rogue security software has been observed (at least) as early as 2005 [28]. More in-depth reports of rogue security software have ranged from analyses on the diffusion of such threats [1], to studies on their social aspects and their comparison to traditional malware economies [18]. Recently, security industry reports [4,8] have presented thorough descriptions of various instances of rogue software, their look and feel as well as some tactics they use. By focusing on a large-scale study of the structure of the distribution infrastructure for rogue AV domains, this work complements previous analyses on rogue security software by offering new lessons on this threat ecosystem and its peculiarities.

We previously provided a preliminary, high-level overview of some of the results obtained with the method described in this paper [8]. The novel contributions of this paper with respect to that technical report are threefold. First, we provide a precise description of the experimental setup and the analysis method.

Second, we give a comparison, thanks to a novel experimental dataset, with other kinds of web-based threats. Third, we supply an ensemble of insights on the economic rationales explaining the identified differences.

Concurrently to our work, Google published a study on the distribution of rogue AV software [21], which focuses on the prevalence of rogue AV software and on its distribution mechanisms. In this paper, we also uncovered the campaigns underlying rogue AV sites and performed an initial study of their victims.

These economic insights contribute at completing the picture on the underground economy and its dynamics. This complements previous works on the topic. Similarly to what is presented here, Moore *et al.* [16] have collected client volume information for a different threat landscape, that of the phishing websites. Holz et al. [11] have instead infiltrated some weakly configured drop-zones to study the extent and the economic aspects of phishing and attack campaigns. Finally, previous work [7,9] has monitored the type of transactions carried out by cyber-criminals through underground channels.

Different techniques have been proposed to study the structure and the diffusion of specific threats. Moshchuk *et al.* [17] have crawled 18 million URLs to quantify the nature and the extent of the spyware problem. Provos *et al.* [19] have analyzed billions of URLs and used a combination of machine learning techniques and crawlers to analyze the infrastructure underneath drive-by downloads. McGrath *et al.* [15] have studied the dynamics associated to the registration of phishing domains. Stone-Gross *et al.* [24] have infiltrated the Torpig botnet and gathered information on its size and dynamics. In all these cases, the authors have used different data collection techniques to generate high-level overviews on specific threats. While this work complements the state of the art by providing an analysis of a previously unexplored threat landscape, that of the rogue security software, our contribution goes beyond that. We show the usefulness of multi-criteria analysis techniques to mine these large datasets and discover specific campaigns within the multitude of domains under observation. We also demonstrate our ability to leverage these techniques to compare different threat landscapes, and identify specific behaviors that are a characteristic of a given threat.

3 Methodology

In this Section, we begin by describing our methodology for collecting information about the rogue AV distribution infrastructure. We then discuss the analysis techniques that we used on the collected data. The data collection itself was carried out over three separate phases: the collection of rogue AV-related domain names, the collection of information on each domain and the discovery of specific campaigns leveraging attack attribution tools developed in the context of the WOMBAT project[5].

[5] `http://www.wombat-project.eu`

3.1 Rogue AV domains

To build an initial seed of domains associated to the rogue AV distribution, we aggregated information from a number of different sources:

- Norton Safeweb (`http://safeweb.norton.com`)
- Malware Domain List (`http://malwaredomainlist.com`)
- Malware URL (`http://www.malwareurl.com`)
- Hosts File (`http://www.hosts-file.net`)

All these sources offer at least a rough categorization of the type of each malicious domain they are listing, and allowed us to systematically collect all the domains that were believed to be correlated to the rogue AV distribution by means of simple heuristics.

To complete our picture on the collected domains, we have integrated our domain list with the information generated by freely accessible IP-NAME mapping datasets (`http://www.robtex.com`). This allowed us to discover all the domain names hosted on each IP where at least one rogue domain had been found.

3.2 Rogue server information

Once the initial list of domains was created, we have collected as much information as possible on each of them, on their relation with the associated web servers, and on their dynamics. To do so, we have taken advantage of HARMUR, a **H**istorical **AR**chive of **M**alicious **UR**Ls also developed in the WOMBAT project.

HARMUR enables us to study the relation between client side threats and the underlying server infrastructure, and their evolution over time. Instead of developing new detection technologies (*e.g.,* based on honeyclients, or special web crawlers), HARMUR integrates multiple information sources and takes advantage of various data feeds that are dedicated to detecting Web threats. By doing so, HARMUR aims at enabling the creation of a "big picture" of the client-side threat landscape and its evolution.

In the specific context of this work, HARMUR generated the necessary contextual information on the identified rogue AV domains, and on all the other domains that were discovered to be sharing the same server as rogue AV domains thanks to DNS mapping information. In order to generate a dynamic perspective on the characteristics of the observed domains, HARMUR implements a set of analysis modules that are re-iterated on each tracked domains on a daily basis:

- Information on the security state of a domain.
  - **Norton Safeweb information.** For each domain, we have queried its security status taking advantage of the Norton Safeweb website reputation service[6]. This allowed us to retrieve information on a variety of threats known to be present on each domain, ranging from browser exploits, to malware samples, to phishing sites.

[6] `http://safeweb.norton.com`

  - **Google Safe Browsing information.** We have taken advantage of the Google Safe Browsing API[7] to detect the presence of threats within a given domain.
- Information on the domain.
  - **Registration information.** We have parsed the registration data obtained via the WHOIS protocol in order to get information on the identity of the registrant and of the provided contact email address, as well as the name of the registrar[8].
  - **DNS relations.** By means of DNS queries, we have retrieved for each domain the associated *NS* records and all the *A* records associated to all the hostnames known to belong to it. Whenever only one domain name was available and we had no information on the associated hostnames, we considered as hostnames the domain name itself and the hostname generated by prepending the standard "www" name.
- Information on the servers.
  - **Geolocation and AS information.** For each web server associated to the rogue domain through a DNS *A* record, we have collected information on its geographical location as well as its associated Autonomous System number.
  - **Server uptime and version string.** By means of HTTP HEAD packets, we have tested the responsiveness of the discovered servers and, by looking at the HTTP response headers, we have collected information on the server configuration by looking at the advertised server version string.

3.3 Limitations

Despite our efforts to maximize the threat coverage by aggregating as many information sources as possible, we are fully aware of the limitations of the dataset at our disposal. Due to the nature of our observational ability and the way the rogue AV ecosystem operates, it is impossible to know with certainty what fraction of the total rogue AV providers across the whole Internet we have been able to observe. For instance, we have noticed a predominance of servers physically located in US. This result might be skewed by the type of heuristics used for identifying rogue AV sites, that could overlook rogue AV servers that are primarily marketed to non-English languages. Moreover the identification of rogue domains is itself a potential source of bias. Our analysis is based on the identification of rogue AV domains performed by third party sources, and does not provide any guarantee in terms or precision of classification. We have

[7] `http://code.google.com/apis/safebrowsing/`

[8] The WHOIS specification [6] requires WHOIS records to be human readable, and does not specify their syntax and their semantics. As a consequence, the data stored by different registrars is often in different formats. We have built a generic parser that handles a vast number of registrars and 17 specific parser for other common registrars, but despite of this effort registration information is not available for all the domains taken into consideration.

indeed identified through manual inspection of our feeds a limited number of domains that did not seem to be actually related to the rogue AV threat type. However, the number of such misclassifications is negligible relative to the size of the dataset. Moreover, when mining the different rogue AV campaigns, any possible pollution of the dataset has been implicitly filtered out by our clustering techniques, as described later.

3.4 Multi-criteria decision analysis

To analyze the campaigns through which rogue AV software is distributed, we have used an *attack attribution* method that relies on a multi-criteria fusion algorithm that has proven to bring several advantages with respect to more traditional clustering methods [25]. Thanks to this method, rogue AV domains are automatically grouped together based upon common elements likely due to the same *root cause*, *i.e.,* same rogue campaign. This attribution method is based on a combination of a graph-based clustering technique with a data aggregation method inspired by multi-criteria decision analysis (MCDA). This method has been successfully used previously to analyze other types of attack phenomena [5,26,27], namely attack events found in honeypot traces.

Generally speaking, the method systematically combines different *viewpoints* such that the behavioral properties of given phenomena are appropriately modeled by the aggregation of all features.

The attribution method used in this paper consists of three components:

1. **Feature selection**: we determine which relevant features we want to include in the overall analysis, and we characterize each element of the dataset according to each extracted feature denoted by $F_k, k = 1, \ldots, n$ (*e.g.,* by creating feature vectors for each element).
2. **Graph-based clustering**: an undirected edge-weighted graph is created regarding every feature F_k, based on an appropriate distance for measuring pairwise similarities.
3. **Multi-criteria aggregation**: we combine the different graphs of features using an *aggregation function* that models the expected behavior of the phenomena under study.

The approach is mostly unsupervised, *i.e.,* it does not rely on a preliminary training phase.

Feature selection. Among the different information tracked through HARMUR, we have selected a number of features that we believed to be likely to reveal the organized operation of one specific individual or group.

- **Registrant email address (F_{Reg})**. Whenever available, the email address provided upon registration of the domain.
- **Web Server IP addresses (F_{IP}), class C ($F_{Cl.C}$), class B ($F_{CL.B}$) subnets**. To allow the identification of servers belonging to the same infrastructure, we have separately considered three features corresponding to the full IP address, its /24 and its /16 network prefix.

- **Nameserver IP address (F_{NS})**. The IP address of the authoritative nameserver(s).
- **Registered domain name (F_{Dom})**. We decided to consider as a feature the domain name itself to be able to detect common naming schemes.

In summary, by analyzing the available features, we have defined the following feature set: $\mathcal{F} = \{F_{Reg}, F_{IP}, F_{Cl.C}, F_{Cl.B}, F_{NS}, F_{Dom}\}$, which will be used by the multi-criteria method to link rogue domains to the same campaign.

Graph-based representation. In the second phase of our attack attribution method, an undirected edge-weighted similarity graph is created regarding each selected feature F_k, based on an appropriate distance for measuring pairwise similarities. A specific definition of similarity had to be defined for each of the considered features.

Since feature vectors defined for F_{IP}, $F_{Cl.C}$, $F_{Cl.B}$ and F_{NS} are simply *sets* of IP addresses (or sets of IP subnets), it is relatively easy to calculate a similarity between two sets by using the *Jaccard similarity* coefficient. This coefficient is commonly used to estimate the amount of overlap between two sets of data.

While simple equality would have been sufficient, we wanted to incorporate into F_{Reg} some additional semantics, taking into consideration the usage of specific email domains or the usage of specific keywords. For this reason, we have given maximum similarity score to identical email addresses, and non-null similarity scores to email addresses sharing same username, same email domain, or both containing popular AV keywords. For the sake of conciseness we refer the interested reader to [25] for more detailed information on this measure.

Finally, we wanted to define a notion of similarity for F_{Dom} able to catch commonalities between rogue domain names having similar patterns, or common sequences of the very same tokens. We have accomplished this goal by using the *Levenshtein distance*[9]. To normalize the Levenshtein distance to a similarity metric, we have used a commonly-used transformation [23] that maps a generic distance value to a similarity score within the interval $[0, 1]$.

Multi-criteria aggregation. As a final step of the multi-criteria analysis, we have used an *aggregation function* that defines how the criteria (*i.e.,* the site features) must be combined to group rogue domains as a function of their common elements.

An aggregation function is formally defined as a function of n arguments ($n > 1$) that maps the (n-dimensional) unit cube onto the unit interval: $f : [0, 1]^n \longrightarrow [0, 1]$. To model complex requirements, such as "most of" or "at least two" criteria to be satisfied in the overall decision function, we have used Yager's *Ordered Weighted Averaging* (OWA) [30].

[9] Levenshtein distance corresponds to the minimum number of operations needed to transform one string into the other (where an operation is an insertion, deletion, or substitution of a single character).

Other possible aggregation functions that allow for more flexible modeling, such as the Choquet integral, may also be used and have been considered elsewhere [25].

Definition 31 (OWA) [2,30] *For a given weighting vector* $\mathbf{w}$*,* $w_i \geq 0$*,* $\sum w_i = 1$*, the* OWA *aggregation function is defined by:*

$$OWA_{\mathbf{W}}(\mathbf{z}) = \sum_{i=1}^{n} w_i z_{(i)} =< \mathbf{w}, \mathbf{z}_{\searrow} > \quad (1)$$

where we use the notation $\mathbf{z}_{\searrow}$ *to represent the vector obtained from* $\mathbf{z}$ *by arranging its components in decreasing order:* $z_{(1)} \geq z_{(2)} \geq \ldots \geq z_{(n)}$*.*

In our application, the vector $\mathbf{z}$ represents the set of similarity values obtained by comparing a given pair of domains with respect to all site features F_k, as defined previously. By associating weights to the *magnitude* of the values rather than their particular inputs, OWA aggregation allows us to define a weighting vector $\mathbf{w}$ that gives lower weights to the two highest scores:

$$\mathbf{w} = [\ 0.10,\ 0.10,\ 0.20,\ 0.30,\ 0.20,\ 0.10\ ]$$

In other words, we assign more importance to features starting from the third highest position. The two highest scores will have lower weights (0.10), and thus *at least three* strong correlations will be needed to have a global score above 0.3 or 0.4, which will be used as a decision threshold to keep a link between two domains. A sensitivity analysis has been performed on this decision threshold to determine appropriate ranges of values [25]; however, due to space constraints, we do not provide further details in this paper.

4 Insights on the rogue security software threat economy

We will now look into the details of the dataset presented in the previous section and try to infer information regarding the modus operandi of the individuals at the root cause of these businesses.

4.1 High-level overview

The dataset at our disposal consists of 6,500 DNS entries, collected between June and August 2009, pointing to 4,305 distinct IP addresses hosting rogue AV servers. At least 45% (2,935) of all domains were registered through only 29 Registrars.

As a first step, we have taken advantage of the DNS information at our disposal to set apart generic hosting services, hosting both rogue AV domains and benign sites, from servers specifically deployed for hosting Rogue AV content.

Version string	# servers
Apache	610
Microsoft-IIS/6.0	218
Apache/2.2.3 (CentOS)	135
Apache/2.2.3 (Red Hat)	123
Apache/2	100
Apache/2.2.11 (Unix) mod_ssl/2.2.11 OpenSSL/0.9.8i DAV/2 mod_auth_passthrough/2.1 mod_bwlimited/1.4 FrontPage/5.0.2.2635	69
Apache/2.0.52 (Red Hat)	49
nginx	33
Apache/2.2.11 (Unix) mod_ssl/2.2.11 OpenSSL/0.9.8e-fips-rhel5 mod_auth_passthrough/2.1 mod_bwlimited/1.4 FrontPage/5.0.2.2635	32
LiteSpeed	26
Others	1498

Table 1. Top 10 server version strings.

Domain	# registered domains
gmail.com	1238 (30.52%)
id-private.com	574 (14.15%)
yahoo.com	533 (13.14%)
whoisprivacyprotect.com	303 (7.47%)
privacyprotect.com	125 (3.08%)
mas2009.com	101 (2.49%)
space.kz	90 (2.22%)
NameCheap.com	85 (2.10%)
domainsbyproxy.com	62 (1.53%)
hotmail.com	59 (1.45%)

Table 2. Top 10 registrant email domains.

We identified all DNS entries resolving to the same IP address, and correlated these with lists of known rogue AV- and malware-serving domains. A total of 2,677 IP addresses (web servers) host only domains that are known to serve rogue AV software. An additional 118 IPs provide services for both rogue-AV and other malware-serving domains. The remaining 1,510 IP addresses host both malicious and benign domains, and are therefore likely to be associated to hosting services unaware of the illicit use of their infrastructure.

Rogue AV servers localization. Mapping the 2,677 IPs hosting only rogue AV software to Autonomous System (AS) numbers, we identified a total of 509 ASes. Interestingly, but yet not surprisingly, the distribution of servers over ASes is skewed towards some specific ASes: approximately 37% (984 servers) are hosted by only 10 particularly popular ASes. As previously pointed out, the geographical distribution of these servers is heavily skewed towards US locations: approximately 53% (1,072 servers) are hosted in the USA.

Rogue AV server versions. When looking at the web server type and version for the 2,677 rogue AV web servers, in some cases we see some very specific configurations that may be indicative of the use of standardized templates or of a single entity/operator using a fixed configuration. Table 1 reports some of the most popular observed version strings. Overall, Apache (in various configurations) seems to be used in well over 40% of the rogue AV servers.

Rogue AV domain registrations. We also looked at the email addresses provided by all Registrants of rogue AV domains. The list of most popular domains, shown in Table 2, contains some of the obvious email hosting services (Gmail, Yahoo! Mail, Lycos, *etc.*). More interestingly, we see that 26% of the analyzed domains make use of anonymous domain registration services such as *domainsbyproxy.com*, *whoisprivacyprotect.com*, *id-private.com*, and *space.kz*. We also see

some cases of ISPs that do not formally offer anonymous domain registration services, but are rather lax in their verification of registrant identities and email addresses. For instance, *Namecheap.com* is often associated to registrant names ranging from "Kyle" to "AA".

Rogue AV domains and browser exploits. While rogue AV software seems to be primarily trying to lure users into downloading software to stop non-existing security issues on their systems (*scareware*), we found it interesting to evaluate the presence of other threats on the domains by correlating them with information provided by web crawlers. We determined that 814 of the rogue AV domains were serving malware of various types; 417 domains attempted to use browser exploits; 12 domains led to the installation of spyware, and 19 domains would cause the installation of a trojan. This result underlines the use, in a minority of cases, of more aggressive strategies for the monetization of clients lured into visiting the domains.

Towards the big picture. Given the size of the dataset, it is beyond the scope of this work to describe all the relationships we discovered in the course of our analysis. We have although tried to generate a "big picture" of the landscape by plotting in Figure 1 the relationships between servers hosting rogue AV content and all the domains known to be hosted on them, a total of 235,086 domains. Due to the complexity of the landscape, we have tried to simplify the visualization by omitting all IPs that were associated to less than 100 different domains. The represented domains comprise both known rogue AV domains and unrelated domains that have been discovered as being hosted on the same server thanks to robtex.com. We have used darker colors to differentiate rogue AV domains from the others.

The subset represented in Figure 1 consists of 174 servers that were hosting a total of 30,632 distinct domain names. In this observed domain set, 15% of the total hosted rogue security software, while 9% were observed to host other types of threats. Interestingly, most of the domain names are linked to a single web server, but some rogue AV domains were associated, over time, to several distinct IP addresses, creating some complex subgraphs such as those in the middle of Figure 1.

Figure 1 shows the complexity of the problem of the identification of malicious domains. It highlights the challenges of protecting the web clients from potentially dangerous sites through IP-based blacklisting methods. Indeed, the coexistence of both rogue and legitimate domains on the same server IP undermine the applicability of such approaches since it would be detrimental to perfectly benign sites. We will explore this issue further in Section 6.

4.2 The campaigns

To get a better understanding of the modus operandi of the rogue AV operators, we have taken advantage of the multi-criteria decision analysis (MCDA)

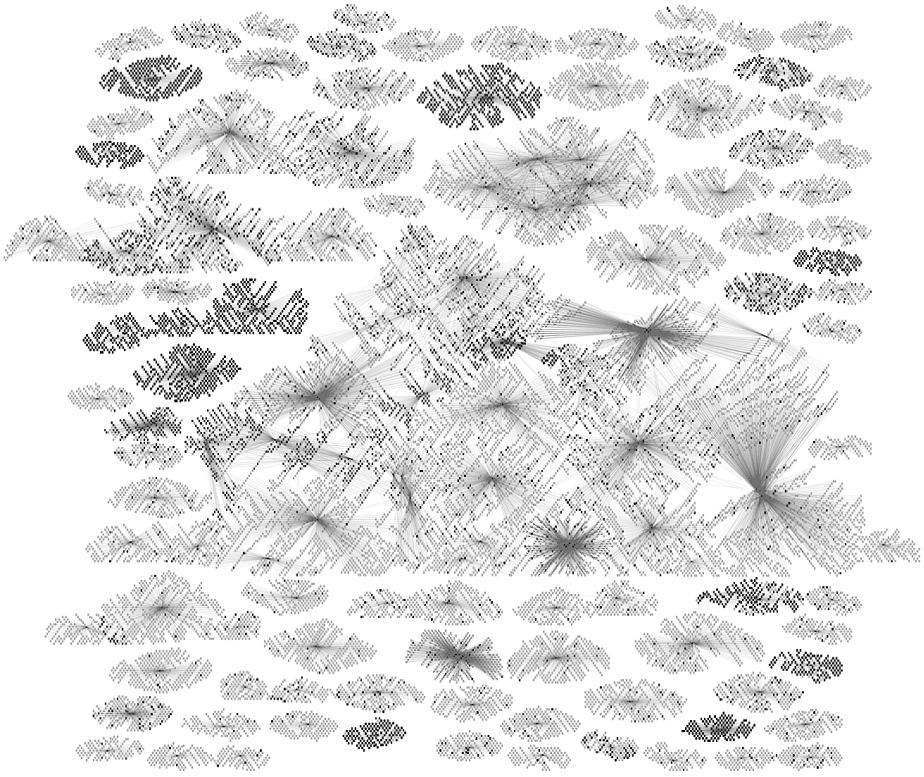

Fig. 1. Relationships between observed domains and the servers hosting them. Darker nodes represent rogue AV domains, while lighter nodes indicate benign domains.

described in Section 3.4 to mine the dataset and identify separate campaigns likely to be generated by the action of a single individual or group.

The application of the method has led to the identification of 127 separate campaigns grouping a total of 4,549 domains. The identified campaigns have an average size of 35.8 domains, but with a significant variance in size. More specifically, 4,049 domains are associated to the 39 biggest campaigns, with the largest comprising 1,529 domains.

In the rest of this Section we will look more closely at three of these campaigns and we will show through their analysis the value of the MCDA in getting insights on the dynamics of the rogue AV threat landscape.

Large-scale campaigns. Some of the campaigns identified by our attribution method consisted of several hundreds domains. One of such examples is represented graphically in Figure 2. The graph represents the relationship between domains (clustered in big, dense groups of small rectangles), the subnets of their hosting servers (represented with larger, lighter rectangles) and the registrant

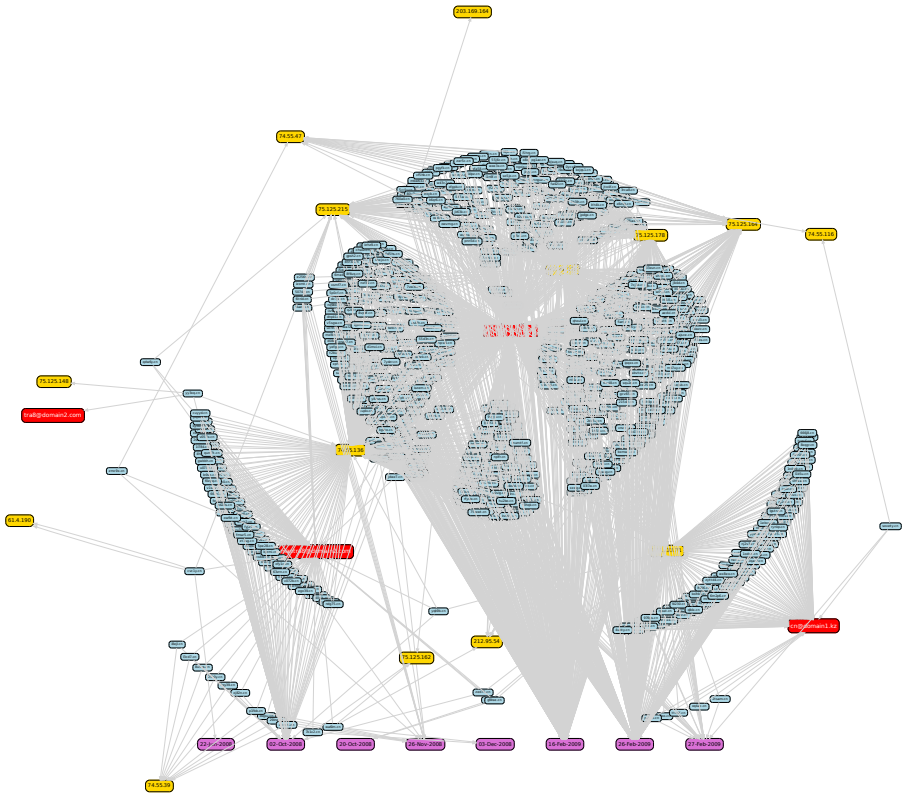

Fig. 2. Graphical representation of a long lasting, larger campaign.

email addresses (represented with large, dark rectangles). The nodes at the bottom of the graph represent instead domain registration dates.

Figure 2 groups about 750 rogue domains that have been registered in the .cn TLD (resolving to 135 IP addresses in 14 subnets), on eight specific dates over a span of eight months. However, despite the apparent link to China, the majority of the IP addresses of the hosting servers were hosted in the United States, Germany, and Belarus. In fact, no server could be identified as being located in China.

Interestingly, the same Chinese registrar (*Era of the Internet Technology*) was used for the registration of all domain names. All of the domain names are composed of exactly 5 alphanumeric characters, apparently chosen in a random fashion (*wxe3x.cn*, *owvmg.cn*,...), which indicates the use of automated tools to create those domains. Another noteworthy characteristic of this campaign is that the registrant responsible for 76% of the domains makes use of a WHOIS

domain privacy protection service (cn@id-private.com), which we have said to be a commonly observed characteristic in certain rogue campaigns.

Finally, a manual analysis of the domains represented in Figure 2 revealed a more complex underlying ecosystem. These domains were in fact linking to a fake scan page hosted on a server belonging to a separate campaign. Such discovery underlines the existence of complex interrelations in this threat ecosystem, interrelations that would have been impossible to discover without the employment of data mining techniques able to reduce a corpus of thousands of domains to few, large campaigns carried out by specific individuals.

PC-Security and PC-Anti-Spyware campaigns. One very good example of such interrelation can be found when looking at two other distinct clusters identified by our approach. The multi-criteria decision algorithm correctly identified them as two distinct campaigns, as they involve different features (different timings, different web servers, *etc.*). However, the analysis of both clusters reveals a common modus operandi used in both cases.

Indeed, the two clusters were composed of a relatively low number of domains (16 and 14) that were clearly referring to anti-virus or anti-spyware "products" (*e.g., pcsecurity-2009.com*, *homeav-2010.com*, *pc-antispyware2010.com*). A number of similarities could be identified in their deployment strategy:

- Both clusters use the exact same domain naming scheme, consisting of the insertion of dashes among a set of fixed words (*e.g., pc-anti-spyware-2010.com*, *pc-anti-spyware-20-10.com*, and *pc-antispyware-2010.com*).
- All of the domains in each cluster use the same registrar (OnlineNIC) and are serviced by the same two ISPs.
- The email addresses of all domain registrants are in ".ru" domains.
- The servers were on consecutive IP addresses, although the two clusters were associated to servers in completely different networks.

Perhaps even more conclusively, upon manual inspection we found that the content of each site was identical, with the exception of one differing image. All this leads us to assume that the deployment of the rogue AV domains was carried out with a good degree of automation by interacting with a single registrar. It is also worth noting that both clusters are split between two different ISPs, suggesting an attempt to provide some level of redundancy in case a cluster is taken offline by the ISP. Finally, we observed that all of the hosting web servers were located in the US. We refer the interested reader to [25] for a more detailed presentation of these two results, as well as other interesting ones.

5 Landscape characteristics

Section 4 provided an in-depth overview of the rogue AV threat landscape, and showed our ability to identify within such landscape articulated campaigns deployed via a certain level of automation. The specificity of these characteristics to the Rogue AV landscape stays although unproved so far. This Section addresses this problem by performing a comparative analysis of the results obtained for the

Rogue AV landscape with those obtained by looking at a completely different web-borne threat: drive-by downloads. We will show that the complexity of the identified campaigns is a very specific characteristic of the rogue AV landscape, and we will go further by looking into the economics of this landscape, showing that the particularly large return on investment largely justifies the complexity of the observed infrastructure.

5.1 Comparison with drive-by downloads

The methodology proposed so far is completely generic, and can be utilized equivalently to study the characteristics of the infrastructure underlying any web-borne threat. We have therefore decided to leverage this fact to compare our findings for the rogue AV threat with those of a specific type of drive-by download. To do so, we have constructed a second dataset taking advantage of data generated by some internal web crawlers and used it as an additional URL feed for HARMUR. Among all the exposed threats, we have chosen to focus on all the landing sites (we use "landing site" as in [19] to denote the site that initiates a drive-by download) that exploited a very specific weakness, namely the *Internet Explorer ADODB.Stream Object File Installation Weakness* (CVE-2006-0003). We have thus repeated the very same experiment as the one performed on the rogue AV dataset, collecting information on the very same network observables and using such information to build domain features for the multi-criteria analysis technique.

While the multi-criteria approach could successfully identify 127 distinct clusters in the rogue AV dataset, 39 of which accounted for more than 60% of the domains, the clustering profile is very different when looking at the browser exploits web sites. Only 15 small clusters have been identified, accounting for only 201 domains (3.8%). This means that the vast majority of domains (96.2%) did not end up in any cluster. In other words, the very same approach that allowed us to identify large correlations within the rogue AV domains seems to be incapable of detecting any significant correlation in the browser exploit landscape. The reason for this striking difference can be found in the different modus operandi associated to these two threat classes. Our methodology aims at identifying correlations indicative of a shared ownership and management of a set of domains. In rogue security software, the infrastructure in charge of luring the victims into installing the products is maintained by the criminals themselves. This includes both the cost of registering the domains and maintaining the hardware, but also the effort of attracting the users towards it.

This does not seem to happen in the drive-by downloads threat landscape: in the vast majority of cases, the landing pages in charge of exploiting the clients are owned and maintained by uncorrelated individuals. As showed also in [19], drive-by downloads mainly operate by compromising legitimate domains that implement weak security practices. What motivates the individuals at the root of the rogue AV threat infrastructure to sustain the additional cost of directly deploying and maintaining these domains? Providing an answer to this question

requires a better understanding on the costs and the revenues associated to the deployment of a rogue AV campaign.

5.2 Rogue AV monetization

Data collection. The problem of studying the victims of online attacks has received much attention in the past few years. The crux of the problem is that attacks and victims are geographically and temporally distributed, and, as a consequence, there is generally no natural vantage point that researchers can leverage to perform their monitoring.

One approach to overcome this problem consists of performing passive measurements of the secondary effects of victims' activity. For example, in the context of spam botnets, researchers have used spam messages [31] and DNS queries [20,22] as proxy indicators of infected machines. Active approaches are also possible. For example, in some cases, researchers have been able to infiltrate part of the attackers' infrastructure, and, thus, gain visibility of its victims directly from "the inside" [12,24].

These are interesting approaches, yet sometimes difficult to implement for practical or legal reasons. Therefore, we decided to use a novel approach to collect information "remotely from the inside".

Indeed, we observed that, in a number of cases, the servers hosting rogue AV sites were configured to collect and make publicly available statistics about their usage. These servers were Apache web servers using the `mod_status` module, which provides a page (by default, reachable at the `/server-status` location) with real-time statistics about the server status, such as the number of workers and the count of bytes exchanged. When the module is configured to generate "extended status" information, it also provides information about the requests being currently processed, in particular, the address of the client issuing a request and the URL that was requested.

We note that the server status data source has a few limitations. In particular, it does not give access to the content of the communications between clients and servers. As a result, we cannot be certain of the outcome of any access: oftentimes, we will see that the same web page (URL) is accessed, without knowing if the access was successful or not. Second, the server status page only provides the IP address of each victim. It is well known that, mostly due to NAT and DHCP effects, IP addresses are only an approximate substitute for host identifiers, and, due to the lack of visibility into the client-server traffic, we cannot use existing techniques to mitigate this problem [3,29]. Despite these limitations, the server status data allows us to gain some visibility into the access behavior of rogue AV clients.

Victim access dataset. In total, we identified 30 servers that provided status information. Of these, 6 also provided information about client requests, which is critical for our analysis. We continuously sampled the server status pages of each of these 6 servers over a period of 44 days and stored the access time, source IP

Rank	Site	Clients (#)
1	windoptimizer.com	55,889
2	inb4ch.com	23,354
3	scan6lux.com	21,963
4	gobackscan.com	19,057
5	pattle.info	14,828
6	goscansnap.com	14,590
7	goscanback.com	11,347
8	tranks.info	10,050
9	cherly.info	9,875
10	phalky.info	9,836

Table 3. Most accessed rogue AV sites.

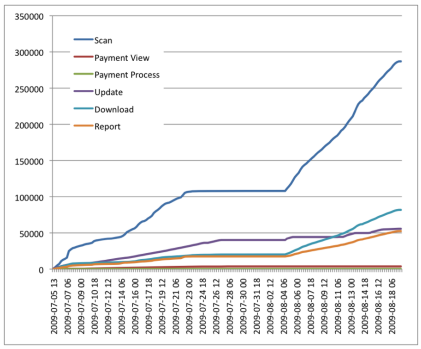

Fig. 4. Cumulative clients activity.

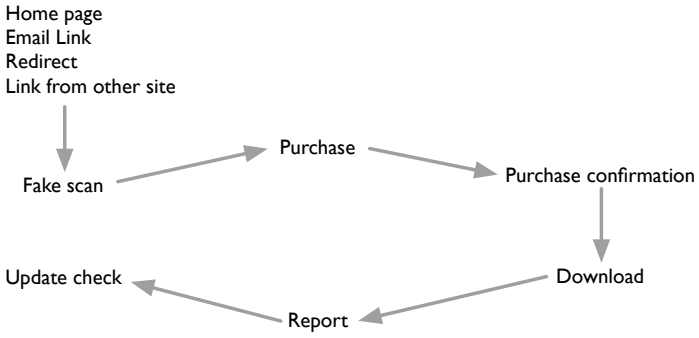

Fig. 3. Typical sequence of accesses by client

address of the client, and the specific URL on the server that was accessed. The 6 servers hosted 193 active rogue AV domains, and an additional 4,031 domains, 62 of which were also rogue AV sites but did not receive any traffic. The remaining 3,969 co-located domains are a mix of malware-serving and benign sites. We then removed from our dataset requests that were not directed at rogue AV sites or that consisted only of probing or scanning attempts. After this filtering, we identified 372,096 distinct client IP addresses that interacted with the rogue AV servers during our observation period.

Localization and server usage. Clients from all around the world interacted with the rogue AV servers. The countries that were most visiting them were USA (147,729 distinct client IPs), UK (20,275), and Italy (12,413). Some rogue AV sites appear to be more popular than others, in terms of the distinct client IP addresses that were served by each. A number of sites only received a handful of clients; in fact, 27 of the rogue AV sites were visited by only 1 client (probably an indication that these sites were no longer, or not yet, actively used). The average number of distinct client IP addresses per rogue AV site was 2,557, with a median of 560 and a standard deviation of 5,405. The 10 most popular rogue AV sites are listed in Table 3.

Access behavior. By clustering the requests issued by the clients according to the requested URL's path, we identified 6 different request types: scan, purchase, purchase confirmation, download, report, and update check. Figure 4 presents the cumulative count of distinct clients (IP addresses) that were observed issuing each type of request. (The presence of the same type of requests on different sites is probably an indication that many rogue AV sites are built on top of the same "rogue AV toolkit.")

As represented in Figure 3, these requests correspond to distinct phases with respect to the interaction of victims with rogue AV sites. A user that is somehow redirected to one of these servers is typically presented with the option to run a *scan* (typically perfunctory) of their computer. The goal of the scan is to scare users into downloading the rogue AV, by reporting that a large number of malware (viruses, Trojans, *etc.*) and other threats are present on their computers. If the scan is successful in this goal, the user will click through to a *purchase* page, or to a "free" download page. In the former case, users enter their information, including payment details (typically a credit card), and are presented with a *purchase confirmation* page. If the charge is successful, or for sites that offer a free "trial" version of rogue AV software, the user is redirected to a *download* page. Once it is successfully installed on the user's computer, the rogue AV software will periodically *check for updates* by querying a specific URL on the server. In certain cases, it will also *report* back to the server the occurrence of specific events (*e.g.,* the download of new updates). During our monitoring, each site handled only a few types of requests. More precisely, a large number of sites were devoted to handling only scan requests, while payment requests were directed at only 7 sites. We speculate that this separation of tasks simplifies the management of the campaign: even when a scan site is taken down, the processing of payments remains unaffected.

Monetization. To determine the success rate of rogue AV servers in convincing clients to download their software, we have counted the number of IPs that had performed, on the same day, a scan followed by a download. We have also counted those that did not perform a download after a scan. Doing so, we observed 25,447 successful and 306,248 unsuccessful scans, leading to the estimation of a 7.7% conversion rate from scan to download.

Similarly, our access data indicates a 1.36% conversion rate from scan to payment. Given an average price for rogue AV software of between $30 and $50, our analysis indicates that these 6 servers (which may be controlled by the same entity, or 6 distinct entities) may generate a gross income of between $111,000 and $186,000 in a period of 44 days. However, this is a best-case scenario; it is likely that at least some of the accesses to the payment URL represented failed or non-existent payments (recall that we do not have access to the actual server response). If we use a more conservative conversion rate between web server access and actual purchase of 0.26%, estimated by others in the context of email spam [13], the gross income for rogue AV site operators in the same period would range between $21,000 and $35,000. The total operational costs

for these rogue AV sites would consist of the cost of hosted web servers and the cost of registering the 193 DNS domains. An informal survey of the providers hosting rogue AV sites indicates that the average monthly cost of a hosted web server is $50. Similarly, the annual domain registration costs vary between $3 and $10. Thus, the costs to the rogue AV operators would range between $1,179 and $2,530 (potentially under $400, if we pro-rate the domain registration for a 44-day period).

While the above cost estimate does not take into consideration the additional cost of advertising the maintained domains through different techniques, the costs are easily covered by the (unknown) income from other illicit activities that piggy-back on the rogue AV distribution flow (*e.g.,* keystroke loggers installed through drive-by downloads by the rogue AV servers).

Ultimately, the easiness with which rogue AV campaigns manage to successfully lure users into purchasing their products generates a return on investment that fully justifies the deployment and the management of complex infrastructures such as those studied in this work.

6 Lessons learned and countermeasures

This work leverages the analysis of real data to study the general characteristics and dynamics of a specific threat landscape, that of rogue security software. We identify the specificities of such threat landscape and their foundations in a particularly favorable market. Such knowledge has direct repercussions on nowadays security practices, and helps underlining weaknesses in currently employed techniques as well as potentials for new research avenues.

Users. Despite of a minor number of cases in which rogue AV domains were observed also in association to other type of threats such as drive-by downloads, the main propagation vector for this type of threat is the psychological impact on the user. The in-depth study of the reasons for the successfulness of the interaction between victims and rogue campaigns is out of the scope of this work, but our analysis clearly shows that users have an important role in the successfulness of rogue AV campaigns. As suggested in [10], the cost-benefit tradeoff associated to the offering of security services is often badly received by the users, that tend to reject the necessity of performing monetary investments to be shielded from hypothetical security threats. Rogue security software leverages this social reality to its own advantage. Increasing user awareness on the cost implicitly associated to security *may* have an impact on the relatively high conversion rates observed in this study, and *may* impact the return on investment associated to rogue AV campaigns.

Blacklisting is strained. Our study revealed two characteristics of the infrastructure used to spread rogue AV that have important consequences on the effectiveness of countermeasures against this threat, and, specifically, of blacklisting, a technique commonly used to prevent end users from accessing malicious resources.

As Figure 1 showed, the rogue AV infrastructure comprises both servers that exclusively host a very large number of rogue AV sites and servers where rogue AV sites coexist with legitimate ones. This situation is a worst case for blacklisting. In fact, IP-based blacklisting (where access to a specific web server IP is blocked) is bound to generate many false positives, thus preventing users from visiting benign sites that happen to be hosted on server IPs that also serve malicious sites. In fact, a naive IP-based blacklisting approach, listing all the servers we identified, would have incorrectly blocked access to 129,476 legitimate web sites. Conversely, domain name-based blacklisting (where access to a specific domain is blocked) is undermined by the easiness with which malicious actors can register large batches of domains. The registration of hundreds of automatically generated domain names observed in the different campaigns is likely to be an active attempt to evade such lists. For example, 77 of the rogue-specific servers that we tracked were associated with more than twenty different domains, with a maximum of 309 domains associated to a single server.

Taking-down rogue AV campaigns. What would be a good strategy then to effectively fight rogue AV campaigns? Our analysis of the victim access dataset hinted at one possible direction: taking down payment processing sites. In fact, these appeared to be less in number than other rogue AV sites (recall that 7 payment sites supported almost 200 front-end "scanning" sites) and seemed to change less frequently. Furthermore, by disrupting the sites generating revenue, defenders are likely to significantly affect also other parts of the rogue AV operations (*e.g.*, registering new sites and paying for hosting).

DNS-based threat detection. This study has highlighted once more the important role of the DNS infrastructure in Internet threats. Rogue AV campaigns often rely on misleading DNS names to lure victims into trusting their products (*e.g.*, *pcsecurity-2009.com*). Also, we have seen how such campaigns often lead to the automated deployment of large numbers of domains pointing to a few servers and following well-defined patterns in their naming schema. For all these reasons, as already noted in [20] for other type of threats, DNS seems to be a promising point of view for the detection of such anomalies.

7 Conclusion

We presented a longitudinal analysis on the infrastructure and the dynamics associated with an increasingly popular threat, that of rogue security software.

The contributions of this paper are threefold. Firstly, we provide the first *quantitative* high-level analysis of the rogue AV threat landscape and the underpinning infrastructure. We detail the relationships between rogue AV domains and the web servers hosting them, and we delve into their characteristics to extract high-level information on the structure of these threats. Secondly, we apply a threat attribution methodology to 6,500 domains under observation and we automatically extract information on large-scale campaigns likely to be associated to the operation of a single individual or group, likely through the help of automated tools. Finally, we provide insights on the economy of the rogue AV

threat landscape by leveraging information on the interaction of victim clients with several rogue AV servers over a period of 44 days. We show how the rogue AV distributors are able to generate considerable revenues through their activities, which fully justifies their investment in the deployment of the distribution infrastructures.

While this paper targets specifically the rogue antivirus threat, we believe that the methodologies and the lessons learnt from our work can be of value to the study of other threats (*e.g.,* phishing and other scams). More specifically, we show how the combination of clustering and data aggregation methods can be leveraged to profile different threat landscapes and, by comparison, offer a valuable tool to the study of threat economies.

References

1. Microsoft Security Intelligence Report, volume 7. Technical report, Microsoft, 2009.
2. G. Beliakov, A. Pradera, and T. Calvo. *Aggregation Functions: A Guide for Practitioners*. Springer, Berlin, New York, 2007.
3. S. Bellovin. A Technique for Counting NATted Hosts. In *Proc. of the Internet Measurement Conference*, 2002.
4. S. P. Correll and L. Corrons. The business of rogueware. Technical report, PandaLabs, July 2009.
5. M. Dacier, V. Pham, and O. Thonnard. The WOMBAT Attack Attribution method: some results. In *5th International Conference on Information Systems Security (ICISS 2009), 14-18 December 2009, Kolkata, India*, Dec 2009.
6. L. Daigle. WHOIS protocol specification. RFC3912, September 2004.
7. M. Fossi, E. Johnson, D. Turner, T. Mack, J. Blackbird, D. McKinney, M. K. Low, T. Adams, M. P. Laucht, and J. Gough. Symantec Report on the Underground Economy. Technical report, Symantec, 2008.
8. M. Fossi, D. Turner, E. Johnson, T. Mack, T. Adams, J. Blackbird, M. K. Low, D. McKinney, M. Dacier, A. Keromytis, C. Leita, M. Cova, J. Overton, and O. Thonnard. Symantec report on rogue security software. Whitepaper, Symantec, October 2009.
9. J. Franklin, V. Paxson, A. Perrig, and S. Savage. An Inquiry into the Nature and Causes of the Wealth of Internet Miscreants. In *Proc. of the ACM Conference on Computer and Communications Security*, 2007.
10. C. Herley. So long, and no thanks for the externalities: the rational rejection of security advice by users. In *Proc. of the 2009 New Security Paradigms Workshop (NSPW)*, pages 133–144. ACM, 2009.
11. T. Holz, M. Engelberth, and F. Freiling. Learning More about the Underground Economy: A Case-Study of Keyloggers and Dropzones. In *Proc. of the European Symposium on Research in Computer Security (ESORICS)*, 2009.
12. T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling. Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm. In *Proc. of the USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2008.
13. C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. Voelker, V. Paxson, and S. Savage. Spamalytics: An Empirical Analysis of Spam Marketing Conversion. In *Proc. of the ACM Conference on Computer and Communications Security*, 2008.
14. B. Krebs. Massive Profits Fueling Rogue Antivirus Market. In *Washington Post*, 2009.

15. K. McGrath and M. Gupta. Behind Phishing: An Examination of Phisher Modi Operandi. In *Proc. of the USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2008.
16. T. Moore and R. Clayton. Examining the Impact of Website Take-down on Phishing. In *Proc. of the APWG eCrime Researchers Summit*, 2007.
17. A. Moshchuk, T. Bragin, S. D. Gribble, and H. M. Levy. A Crawler-based Study of Spyware on the Web. In *Network and Distributed System Security Symposium*, pages 17–33, 2006.
18. H. O'Dea. The Modern Rogue — Malware With a Face. In *Proc. of the Virus Bulletin Conference*, 2009.
19. N. Provos, P. Mavrommatis, M. Rajab, and F. Monrose. All Your iFRAMEs Point to Us. In *Proc. of the USENIX Security Symposium*, 2008.
20. M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A Multifaceted Approach to Understanding the Botnet Phenomenon. In *Proc. of the Internet Measurement Conference*, 2006.
21. M. A. Rajab, L. Ballard, P. Mavrommatis, N. Provos, and X. Zhao. The Nocebo Effect on the Web: An Analysis of Fake Anti-Virus Distribution. In *Proc. of the USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2010.
22. A. Ramachandran, N. Feamster, and D. Dagon. Revealing Botnet Membership Using DNSBL Counter-Intelligence. In *Proc. of the Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, 2006.
23. R. N. Shepard. Multidimensional scaling, tree fitting, and clustering. *Science*, 210:390–398, 1980.
24. B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your Botnet is My Botnet: Analysis of a Botnet Takeover. In *Proc. of the ACM Conference on Computer and Communications Security*, 2009.
25. O. Thonnard. *A multi-criteria clustering approach to support attack attribution in cyberspace.* PhD thesis, École Doctorale d'Informatique, Télécommunications et Électronique de Paris, March 2010.
26. O. Thonnard, W. Mees, and M. Dacier. Addressing the attack attribution problem using knowledge discovery and multi-criteria fuzzy decision-making. In *KDD'09, 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Workshop on CyberSecurity and Intelligence Informatics, June 28th - July 1st, 2009, Paris, France*, Dec 2009.
27. O. Thonnard, W. Mees, and M. Dacier. Behavioral Analysis of Zombie Armies. In C. Czossek and K. Geers, editors, *The Virtual Battlefield: Perspectives on Cyber Warfare*, volume 3 of *Cryptology and Information Security Series*, pages 191–210, Amsterdam, The Netherlands, 2009. IOS Press.
28. Y.-M. Wang, D. Beck, X. Jiang, and R. Roussev. Automated Web Patrol with Strider HoneyMonkeys. Technical Report MSR-TR-2005-72, Microsoft Research, 2005.
29. Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber. How Dynamic are IP Addresses? In *Proc. of the Conference of the ACM Special Interest Group on Data Communication (SIGCOMM)*, 2007.
30. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision-making. *IEEE Trans. Syst. Man Cybern.*, 18(1):183–190, 1988.
31. L. Zhuang, J. Dunagan, D. Simon, H. Wang, I. Osipkov, G. Hulten, and J. Tygar. Characterizing Botnets from Email Spam Records. In *Proc. of the USENIX Workshop on Large-Scale Exploits and Emergent Threats*, 2008.