Gone Rogue: An Analysis of Rogue Security Software Campaigns

(Invited Paper)

Marco Cova[*], Corrado Leita[†], Olivier Thonnard[‡], Angelos Keromytis[†] and Marc Dacier[†]
[*]*University of California, Santa Barbara, marco@cs.ucsb.edu*
[†]*Symantec Research Labs, {Corrado_Leita,Angelos_Keromytis,Marc_Dacier}@symantec.com*
[‡]*Royal Military Academy, Belgium, olivier.thonnard@rma.ac.be*

Abstract

In the past few years, Internet miscreants have developed a number of techniques to defraud and make a hefty profit out of their unsuspecting victims. A troubling, recent example of this trend is cyber-criminals distributing rogue security software, that is malicious programs that, by pretending to be legitimate security tools (e.g., anti-virus or anti-spyware), deceive users into paying a substantial amount of money in exchange for little or no protection.

While the technical and economical aspects of rogue security software (e.g., its distribution and monetization mechanisms) are relatively well-understood, much less is known about the campaigns *through which this type of malware is distributed, that is what are the underlying techniques and coordinated efforts employed by cyber-criminals to spread their malware.*

In this paper, we present the techniques we used to analyze rogue security software campaigns, with an emphasis on the infrastructure employed in the campaign and the life-cycle of the clients that they infect.

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 actually provides the user with little or no protection. In some cases, rogue security software (in the following, more compactly *rogue AV*) actually facilitates the installation of the very malicious code that it purports to protect against.

Rogue AV makes its way on victim machines in two prevalent ways. First, social engineering techniques can be used to convince unexperienced users that a rogue tool is legitimate and that its use is necessary to remediate often inexistent or exaggerated threats found on the victim's computer. A second, more stealthy technique consists of attracting victims on 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.

Rogue AV programs are distributed by cyber-criminals to generate a financial profit. In fact, after the initial infection, victims are typically tricked into paying for additional tools or services (e.g., to upgrade to the full version of the program or to subscribe to an update mechanism), which are most often fictitious or completely ineffective. The cost of these scams range from $30–$100.

Despite its reliance on traditional and relatively unsophisticated techniques, rogue AV has emerged as a major security threat, in terms of the size of the affected population (Symantec's sensors alone reported 43 million installation attempts over a one-year period [1]), the number of different variants unleashed in-the-wild (over 250 distinct families of rogue tools have been classified [1]), and the volume of profits generated for cyber-criminals (upward of $300,000 a month in affiliate commissions alone [2]).

The prevalence and effectiveness of this threat has spurred considerable research by the security community [1], [3], [4]. These studies have led to a better understanding of the technical characteristics of this malware (e.g., its advertising and installation techniques) and of the quantitative aspects of the overall threat (e.g., the number and geolocation of the web sites involved in the distribution of rogue programs and of their victims).

However, a number of areas have not been fully explored. In particular, malware code, the infrastructure used to distribute it, and the victims that encounter it do not exist in isolation, but are different aspects of the coordinated effort by cyber-criminals to spread rogue AV. We refer to such a coordinated activity as a *campaign*. In our work, rather than examining or measuring single aspects of a rogue AV campaign, we analyzed the campaign as a whole, focusing, in particular, on understanding its infrastructure (e.g., the web servers, DNS servers, and web sites it uses) and the way it is created and managed; and on how it affects the clients that interact with it. More precisely, the main contributions of our work are:

- We developed a methodology to identify the server components used in a rogue security software campaign and to learn any emerging patterns in the ways these are set up and organized.
- We leveraged the results of our analysis to provide insights into the tools, techniques, and strategies followed by current campaigns.
- Finally, we have performed an initial analysis of the behavior and life-cycle of infected clients, as observed from within the infrastructure responsible for the infection (as opposed to the client end-point).

2. Our Approach

Server-side analysis. A primary goal of our analysis is to identify the server-side components involved in a campaign. While extensive lists of individual rogue AV-hosting sites are easily obtainable from telemetry data of legitimate anti-virus tools or publicly-available blacklists, this data *per se* does not provide information on the campaigns themselves, for example, whether two sites are part of the same campaign.

One assumption that can reasonably be made is that a campaign is managed by a group of people, who are likely to reuse, at various stages of the campaign, the same techniques, strategies, and tools. Consequently, our approach to studying the infrastructure of rogue AV campaigns (e.g., rogue AV-hosting sites, DNS servers) is based on identifying commonalities in the sites employed in the campaigns. More precisely, we apply multi-criteria clustering to determine groupings of server components with similar characteristics (the interested reader can refer to [5] for the details of our clustering techniques). The features we used for the clustering consisted of a number of “network observables” including IP address(es), DNS names, other DNS entries pointing to the same IP, geolocation information, server identification string and version number, ISP identity, AS number, DNS registrar, DNS registrant, and server uptime. Values for each of these features were collected over a period of approximately two months. The rogue AV-hosting servers that we analyzed using these techniques were identified through a variety of means, including automated and manual feeds. In total, we considered 4,305 rogue AV-hosting IP addresses with 6,500 related domain names.

Our analysis identified 39 distinct clusters comprising 10 or more domains, which we interpret as different campaigns. Figure 1 shows one such cluster, which comprises 15 distinct rogue AV-hosting sites (blue nodes). A number of details support the hypothesis that these sites are indeed part of the same campaign. First, registration data shows that all sites were registered on the same day using only 3 email addresses from `.ru` domains (red nodes). Second, all sites have been hosted at one point in time in the same AS, even more strikingly on consecutive IP addresses (yellow nodes). Furthermore, the site names follow the same naming scheme, which consists of different combinations of basic tokens such as `home`, `av`, `anti-virus`. Finally, even more conclusively, we found that the content of each site was identical (notice that the site content is not a feature of our clustering system).

Other clusters represent more sophisticated campaigns. For example, our analysis isolated a cluster (not discussed here in detail for sake of space) describing a campaign involving 750 web sites, registered over a period of 8 months, and hosted in several distinct ASes and countries.

Client-side analysis. During our study we found that 6 of the rogue AV-hosting servers we monitored were leaking information about the clients accessing them and their requests. The data available to us during this monitoring was limited to the access time, the IP address of the client, and the specific URL on the server that was accessed. In particular, we did not have access to the content of the client-server communication, for example, we could not be certain whether client requests were successful or not. Despite these limitations, this data provides an interesting view of the (potential and actual) victims of a campaign, as seen from inside the rogue AV infrastructure. We report here some of the results from our 45-day monitoring, during which we observed 372,096 distinct client IP addresses.

Rogue AV victims can issue several different types of requests, depending on their current stage in the infection. For example, *scan* requests cause a fake scan of the victim's computer to be displayed. These requests are typical of the before-the-infection phase, when the victim should be scared into downloading and installing the rogue AV. Other requests are related to the installation and use of the rogue AV: *download* requests retrieve the actual binary, *update* requests check if new versions of the rogue AV or of its (perfunctory) virus definitions are available. Other classes of requests we observed include accesses to pages that present a *payment form* (where victims pay for the rogue AV) and *payment confirmation* pages. Finally,

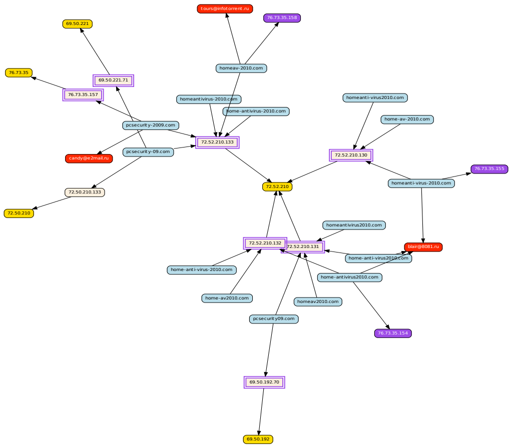

Figure 1. A relatively simple rogue AV campaign.

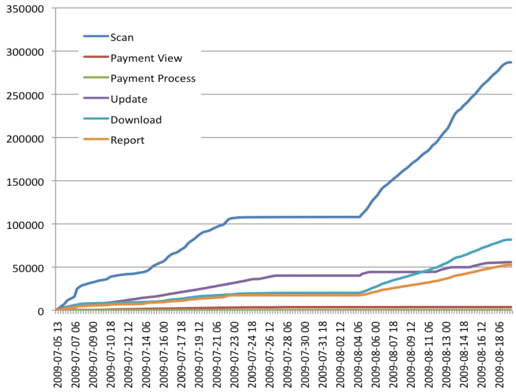

Figure 2. Duration of client interactions (by /24 subnet).

report requests relay to the server various events (e.g., a successful installation).

Most of the clients interacted with any of the servers we monitored only for a few days. More precisely, we noticed that less than 10% of the clients connected to one of the servers 15 days after their first visit (for this measurement, we identify a client by its /24 subnet address to offset DHCP and NAT effects). This is a surprising result as some of the request types described above (e.g., the update requests) would suggest long-lasting interactions. We speculate that this may indicate that the majority of clients do not fall for the scare techniques (thus avoiding the infection) or are quick at removing a rogue AV after it is installed. Alternatively, this result may be an artifact of our limited visibility of client requests, if, for example, clients successfully infected by rogue AV communicated with a different set of servers.

3. Conclusions

We analyzed rogue AV campaigns by analyzing server-side and client-side data. Our clustering technique used various server features to identify 39 campaigns out of 6,500 involved sites. These results can be leveraged in several ways. First, they give a more explanatory description of the rogue AV threat, in which, for example, individual, disconnected sites are substituted by sets of related sites and time relationships (e.g., dates of domain registrations) are explicit. Second, campaign-level information reveals the modus operandi of the criminals orchestrating the campaign, for example, their registration and hosting partners, the duration of their efforts, the sophistication of the tools available to them (e.g., to automate the registration of domain names), and the countermeasures they employ against take-down efforts (e.g., moving web sites from one IP address to a different, fresh one). Finally, the patterns discovered by our clustering analysis could yield means for identifying additional rogue AV sites proactively or reactively. Analysis of client-side data shed some light into the life-cycle of potential and actual victims of rogue AV and their interactions with rogue AV sites. The results of these analysis may be used to derive models of the network traffic of infected machines or to estimate the resilience of rogue AV to cleanup efforts.

Acknowledgment. Marco Cova is a 2009 Symantec Research Labs Graduate Fellowship Winner and contributed to this work while spending his internship within Symantec Research Labs in France. 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.

References

[1] Symantec, "Rogue Security Software," http://www.symantec.com/business/theme.jsp?themeid=threatreport, Tech. Rep., 2009.

[2] B. Krebs, "Massive Profits Fueling Rogue Antivirus Market," in *Washington Post*, 2009.

[3] Microsoft, "Security Intelligence Report (SIR)," http://www.microsoft.com/security/portal/Threat/SIR.aspx, Tech. Rep., 2008.

[4] R. Sherstobitoff and S.-P. Correll, "Rogue Security Software in 2008," *ISSA Journal*, Jan 2009.

[5] O. Thonnard, W. Mees, and M. Dacier, "Addressing the Attack Attribution Problem Using Knowledge Discovery and Multi-Criteria Fuzzy Decision-Making," in *Proceedings of CSI-KDD*, 2009.