BURN: Baring Unknown Rogue Networks

Francesco Roveta
francesco.roveta@mail.polimi.it
Politecnico di Milano

Luca Di Mario
luca.dimario@mail.polimi.it
Politecnico di Milano

Federico Maggi
fmaggi@elet.polimi.it
Politecnico di Milano

Giorgio Caviglia
giorgio.caviglia@polimi.it
Politecnico di Milano

Stefano Zanero
zanero@elet.polimi.it
Politecnico di Milano

Paolo Ciuccarelli
paolo.ciuccarelli@polimi.it
Politecnico di Milano

ABSTRACT

Manual analysis of security-related events is still a necessity to investigate non-trivial cyber attacks. This task is particularly hard when the events involve slow, stealthy and large-scale activities typical of the modern cybercriminals' strategy. In this regard, visualization tools can effectively help analysts in their investigations. In this paper, we present BURN, an interactive visualization tool for displaying autonomous systems exhibiting rogue activity that helps at finding misbehaving networks through visual and interactive exploration. Up to seven values are displayed in a single visual element, while avoiding cumbersome and confusing maps. To this end, animations and alpha channels are leveraged to create simple views that highlight relevant activity patterns. In addition, BURN incorporates a simple algorithm to identify migrations of nefarious services across autonomous systems, which can support, for instance, root-cause analysis and law enforcement investigations.

1. INTRODUCTION

Despite the vast plethora of production-ready, automatic tools to detect suspicious or malicious activity on the Internet (e.g., phishing, spamming, botnet traffic), manual analyses conducted by security experts are still fundamental to investigate non-trivial attacks. This task, however, is particularly hard when slow, stealthy and large-scale activities are involved. To ease manual inspection of security-related events, visualization tools are leveraged to help analysts in their investigations [4]. Often, these tools include heat-maps, or other geo-referenced screens, where security events are pinpointed with symbols of different sizes and colors. Although geo-referenced displays help at spotting countries or areas characterized by the activity of interest, they do not effectively aid analysis at different scales. Indeed, as noted in [8], geo-referenced maps often produce cluttered, confusing maps because of lack of normalization. For example, areas with faster and more developed Internet connectivity will look "dense", yet they will not necessarily reflect the actual magnitude of suspicious activity.

We propose BURN, short for *Baring Unknown Rogue Networks*, an interactive visualization tool for analyzing autonomous systems (ASs), that are groups of networks controlled by the same organization, that are characterized by suspicious activity. As previous research have done, most notably the FIRE system presented in [7], concentrating on ASs is important because they constitute the smallest authority (e.g., an ISP) that manages a certain address space on the Internet. Therefore, it is usually doable during law enforcement or security investigations to contact the people behind ASs if needed; on the other hand, it is normally difficult to reach who is behind, for example, a single internet address. BURN is a free public online tool—currently in private alpha[1]—available both to security experts and end users. Besides its simplicity, the key features of our approach are interactivity, information richness, and dynamicity. BURN supports security analysts to find misbehaving networks, primarily through visual and interactive exploration of time series of events. We integrate up to seven dimensions in a single visual element, which displays four visual variables, thus avoiding cumbersome and confusing maps: By appropriate use of size, position, shape and color, features such as number of malicious servers, their geographical location, AS size, the different types of malicious activities that take place, and the overall AS "rogueness", can be all integrated in a single, yet lightweight screen. In addition, we leverage animations and alpha channels to highlight relevant instances of the malicious behavior of interest. Finally, BURN incorporates a simple algorithm to identify migrations of nefarious services across ASs. These occur, for instance, when an AS blocks some IPs, involved suspicious activity, and the miscreants transport said services to more "friendly" ASs. BURN visualizes these migrations for simple manual inspection.

BURN implements the visualization-as-a-transformation-process paradigm, recently proposed by Masud and collaborators in [5], which highlights how visualization tools are always part of a communication and understanding process, where the user experience and the context are key drivers of the design process. The exploration of rogue ASs in BURN is indeed supported by multiple views that, on the one hand, allow the analyst to observe malicious activity from multiple angles, to browse backward and forward in time, through many levels of details, and, on the other hand, to concentrate on the goal of each work session.

In summary, this paper makes the following contributions:

- We propose a visualization system to display and, more importantly, explore temporal data about rogue ASs. Its visualizations make it easy for security experts to quickly spot malicious events. The motivations behind our system are discussed in §2 after an overview of related work, while a high-level description of the system is provided in §3.

- We exploit basic animation techniques and alpha channels to create plots of malicious events. Our approach helps at displaying more than four variables at the same time, a fairly common situation in security scenarios, without cluttering the screen. When applied to real-world data, this technique allows to visually highlight particularly-malicious ASs, making it easy to recognize relevant ones. This and other details are described in §4.

[1] `http://burn.vplab.elet.polimi.it/`

- We propose a visualization of malicious service migrations across ASs. The algorithm that detects migrations, described in §4.1.1, looks for sudden drops in the number of malicious hosts in one AS that are followed by corresponding subsequent sudden increases in the number of similar malicious hosts in another AS.

2. BACKGROUND AND RELATED WORK

Visualization of malicious hosts and networks on the Internet is useful for law enforcement, cyber-crime investigations, security research, and for investigations requiring manual inspection of security data. For example, the FIRE system, proposed by Stone-Gross and colleagues in [7], automatically finds rogue ASs that host several machines engaged in sustained malicious activity. FIRE, which is short for *FInding Rogue nEtworks*, flags ASs as rogue based on a score that summarizes the amount of events that suggest the presence of hosts engaged in phishing, spamming, hosting drive-by download malware, or botnet traffic. Such events come both publicly-available data (e.g., PhishTank) and network traffic collected by running bots in controlled environments. A similar approach is applied in EMBER, proposed by Yu and collaborators in [8], which visualizes, on a per-city basis, security incidents drawn from the DShield database. NICTER, proposed by Innue et al. in [2], takes a different direction: It leverages animations to create live views of packet flows across Internet nodes. Even if NICTER is mostly network-centric, it is useful to spot spikes of packet flows, which may be insightful to spot attacks.

Approximate geographical location of hosts on the Internet is trivial to obtain nowadays. As a result, geographical displays are widely used in security visualizations: Hosts are pinpointed on a worldwide map with symbols of different sizes and colors, depending, for instance, on the magnitude of the events that are visualized. This approach, however, has two main drawbacks. First, it produces cluttered maps where truly relevant areas are hard to spot. For example, areas with dense populations and large amounts of Internet-connected hosts are often highlighted in said maps, just because they have high chances of being characterized by malicious activities. Second, geo-referenced maps alone are of little help at performing thorough analysis of malicious activities. For example, the FIRE global map represents ASs with pins carrying information such as the IPs of machines hosting malicious services. This information alone is useful to authorities but is of little help to analysts, who would then need to correlate it with other data (e.g., time series of malicious events, security reports) to produce more meaningful results.

EMBER addresses the former drawback with a normalization technique, which accounts for the estimated size (in terms of unique IPs) of city areas; FIRE adopts a very similar mechanism. Dashboards such as the one presented recently by Harrison and coworkers in [1] may alleviate the latter drawback: Multiple plots and graphs (e.g., spectral, temporal, links between hosts as interconnected nodes) coexist on the screen and show malicious behaviors from multiple perspective. EMBER leverages similar dashboards to display the ranking of malicious IPs together with a geographical map, plus histograms of the empirical distribution of various features such as number of malicious IPs. Another notable example is TrGeo [3], which uses a live dashboard that includes a Google Map, a scatter plot and a pie chart that shows the origin, absolute number and categories, respectively, of attacks detected by honeypot sensors. On the one hand, dashboards provide a quick glance over the difference facets of the observed phenomenon. On the other hand, however, such displays may be confusing, because the user is has to keep track of the many objects on the screen.

In addition to the aforementioned research approaches, a wide variety of web-based global threat monitoring tools exist with the main purpose of promoting security-awareness among end users. A notable, recent example is the Norton Cybercrime Index by Symantec[2], a set of dashboards that display trends for each type of threats (e.g., spam, identity fraud, malware, phishing). Also, the Australian Honeynet Project consortium released a dynamic, geo-referenced map to visualize time series of spam-sending events[3], displayed as red dots. This tool is now incorporated in production-ready tools by Clarified Networks[4]. Another example is Akamai's Real-Time Web Monitor[5], a live, geo-referenced worldwide heat-map to visualize Internet connectivity speed, latency and attacks detected in each region. Although these tools are only marginally relevant to the research community, we argue that they play an important role to the increase of awareness about Internet threats. Moreover, because these tools are typically designed for inexperienced users, their interface is particularly intuitive and, in general, they are very easy to use. BURN pursues both the objectives and targets both end users and experts.

3. SYSTEM OVERVIEW

BURN provides a first visualization layer, called *global view*, which overviews rogue ASs, and a detailed visualization layer, *autonomous system view*, which displays detailed information about each AS. The global view is meant for end users, whereas the autonomous system view is useful to researchers and practitioners, for exploring the malicious activity of ASs in depth. Although global view and AS view are loosely coupled together, so that the user is not constrained with mandatory steps throughout the analysis process, a lightweight bridging mechanism between the two is implemented to support the analysis: In the global view, the user can add interesting ASs to the *autonomous system tracking list*—detailed in §4.2.4—then switch across different visualizations and observe, from time to time, how tracked ASs behave, without losing the focus of the analysis.

3.1 Global view

The global view comprises a bubble chart, a geographical map and a trend chart. In the *bubble chart*, ASs, visualized as bubbles, are grouped together with respect to the malicious score (which represents a degree of "rogueness" of the AS, as detailed in §4). The *geographical map* shows the distribution of the ASs on a worldwide and countrywide scale. The *trend chart* visualizes time evolution of ASs, and allows identification of phenomena over time. Three buttons, at the top-center area of the screen, allow to switch between the three visualizations.

The user controls the visualized data by means of a timeline and three filters (detailed in §4.2.1). Both bubble chart and geographical map include the interactive, auto-hiding *timeline* at the bottom of the screen for quick time range selection, as detailed in §4.2.2. Whenever the time range is changed, charts are updated. The user can also use the *activity filter* and *country filter*, both omnipresent, to slice the current chart by activity type (i.e., spam, phishing, malware hosting, botnet traffic) and by country, respectively.

The bubble chart and geographical map provide an additional filter, called *highlight filter*, that highlights ASs exhibiting a malicious activity of interest, chosen from a drop-down list. In addition to the

[2] http://us.norton.com/?cci=on&s_tnt=22618:13:0

[3] http://honeynet.org.au/?q=time_series_geomapping_of_spam

[4] https://www.clarifiednetworks.com/ClarifiedVisualizationGallery#Situation_Rooms_-_Intuitive_views

[5] http://www.akamai.com/html/technology/dataviz1

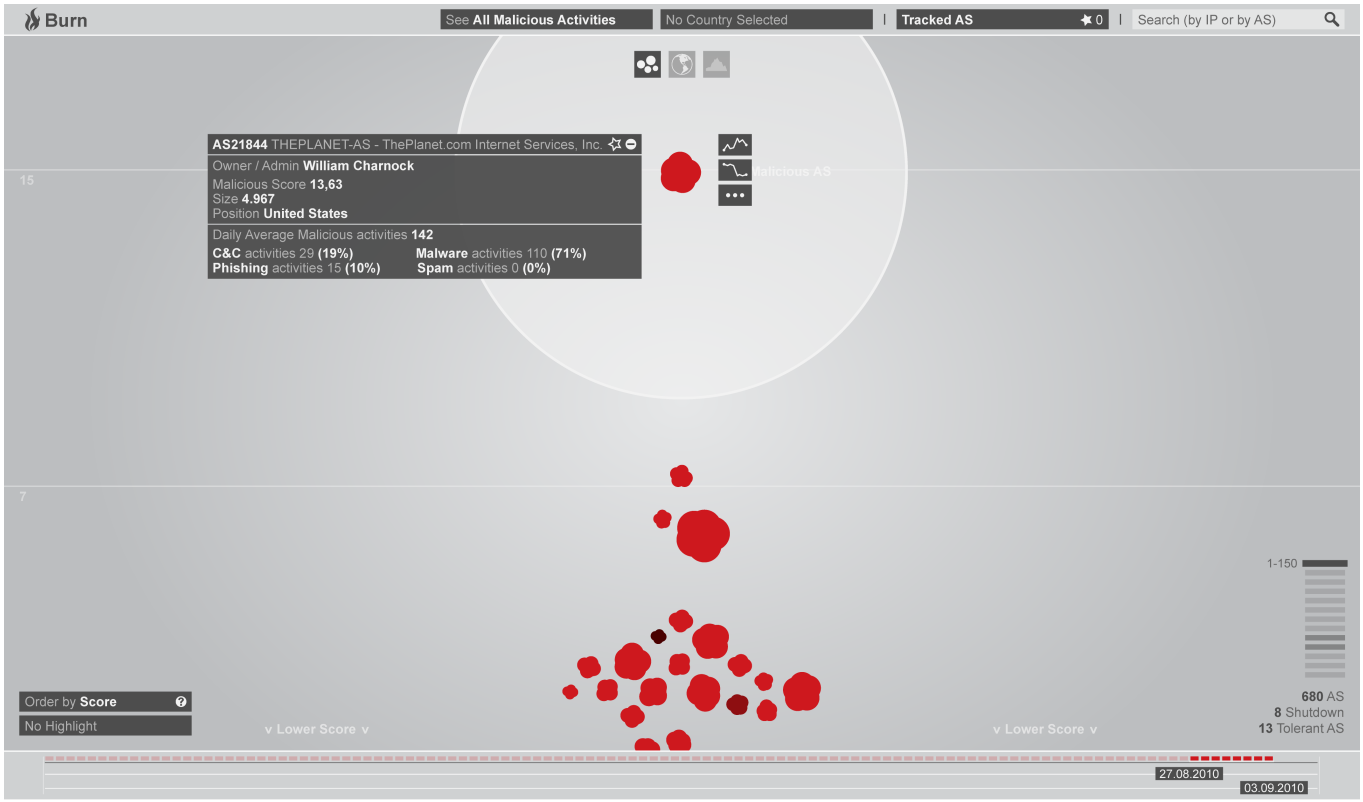


Figure 1: The main screen of BURN is the bubble chart of the global view (defined in §3.1). Autonomous systems are represented as animated bubbles, colored proportionally on their degree of "rogueness". Bubbles are sorted on the vertical axis such that the most relevant autonomous systems appear at the top. As detailed in § 4.2.5, the combination of color shades and animations allows to spot relevant behaviors immediately, while the vertical sorting allows to keep focus on the most rogue autonomous systems, which appear always at the top.

above filters, an omnipresent search box at top-right corner of the screen allows to narrow the analysis around a given IP or AS number.

3.1.1 *Bubble chart*

The bubble chart is the main screen of BURN and represents ASs as bubbles parametrized with respect to two main variables at the same time: number of malicious servers, represented by the area of each bubble, and malicious score, encoded by the vertical position of each bubble. The user can change the variable projected on the vertical axis by choosing between malicious score, which is the default, increments, variation of the malicious score calculated on a daily basis, and size, an estimation calculated by FIRE of the number of all the machines connected to an AS. Relevant characteristics of ASs are visualized by means of the simple animations detailed in §4.2.5.

We use a bubble chart as the main view so that to give an quick glance over the most malicious ASs and, at the same time, provide an immediate comparison of the maliciousness level of the ASs. Although this could be accomplished out also through other visualization techniques (e.g., barcharts) we opted for the bubble chart, because circle shapes let us to better apply further effects (i.e., animations), which are hard to integrate into other layouts. Moreover, circle shapes provide a more efficient use of the space as well as an intuitive way to create visual clusters.

As exemplified in the screenshot in Fig. 1, unlike common bubble charts, the horizontal axis has no variables associated to it. In fact, BURN automatically determines the horizontal positioning of each bubble by optimizing the space between the bubbles and avoiding overlaps. More precisely, we place ASs with similar values close to each other to create visual clusters, which are intuitively associated to "groups" of ASs that share common characteristics. Because ASs with higher values appear at the top of the visualization, relevant ASs are immediately recognizable.

The user can skim through all the ASs in the database by vertical scrolling. BURN aids this with a chart navigator and summary counters of the elements visualized on the chart, which are rendered at the bottom-right corner of the screen.

3.1.2 *Geographical map*

The geographical map shows the distribution of the ASs through two discrete zoom levels: worldwide and countrywide. Unlike other approaches, which often adopt continuous zoom on maps, we purposely opt for two zoom levels, each with a different visualization model, so that we avoid cluttered and confusing maps: At worldwide level, ASs are grouped by country, leading to a clear and synoptic view, whereas the ASs' detailed positions and information are visible only countrywide.

As exemplified in Fig. 2, the *worldwide* geographical map is a single-color choropleth map (see pages 380–402 of [6]), in which the color saturation is calculated as the number of malicious ASs, normalized by the total number of ASs in each country. The user can choose to rearrange the map to display the average number of shutdowns or the average degree of tolerance of the ASs in each country (described in §4.1). Summary information (e.g., number of malicious ASs) appears when hovering over the country area.

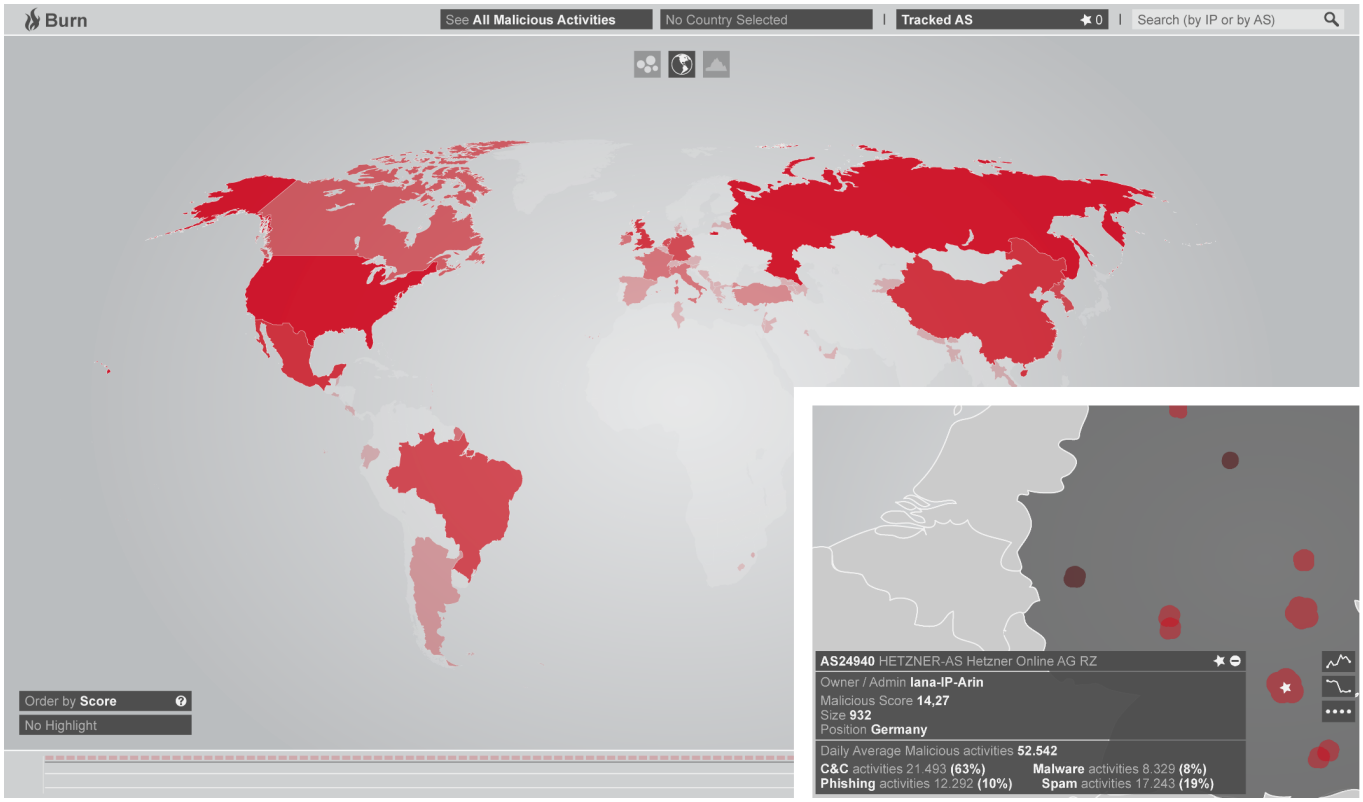


Figure 2: The worldwide geographical map is a single-color, choropleth map where the saturation reflects the average malicious score of each country's ASs. A cropped screenshot of the countrywide map is shown along with an expanded contextual dialog, which shows details about the selected AS (e.g., country, malicious score, activity type, size).

The *countrywide* geographical map, triggered by clicking on the country area, focuses on a single country and shows the ASs' geographical locations on a thematic map (see pages 493–526 of [6]). Each AS is represented by an animated bubble, with exactly the same graphical characteristics displayed in the bubble chart. Notably, we purposely choose a low-resolution map. The reason is because geo-localization of the area covered by the AS is obviously imprecise; therefore, accurate pinpointing could be misleading, especially for inexperienced users.

3.1.3 Trend chart

The trend chart, exemplified in Fig. 3, displays the yearly trends of the number of malicious servers worldwide. We implemented it as a multi-line plot, which shows the breakdown of the number of malicious servers over time, with one line per activity type (i.e., spam, phishing, malware hosting, botnet traffic), plus an additional line for the overall sum. The user can browse the dataset back and forth, year by year, through a set of controls positioned at the bottom of the screen.

This chart is useful to overview the global situation and spot activity peaks. When hovering over each line, additional details on the selected activity and point in time are displayed, whereas the other lines change their opacity to allow a better visualization.

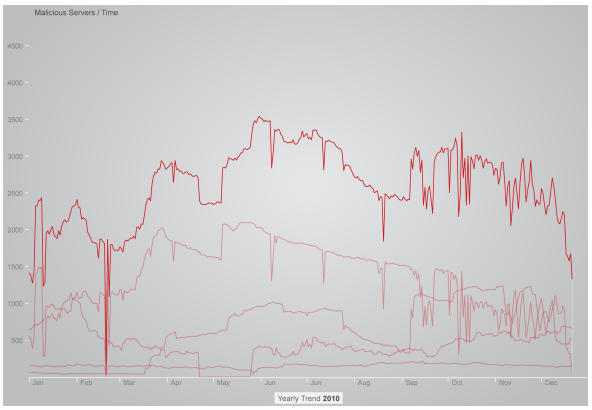


Figure 3: A screenshot of the trend chart of 2010, which shows the breakdown of the number of malicious servers over time, with one line per activity type (i.e., spam, phishing, malware hosting, botnet traffic), plus an additional line for the overall sum.

3.2 Autonomous system view

The autonomous system view is divided into three parts. The *history chart* plots the daily amount of malicious hosts in the selected AS and the average AS malicious score, the *service migration screen* displays the time ranges with substantial migrations of services over other ASs, and the *service longevity chart* highlights long-living IPs (i.e., hosts) against which the AS takes no or ineffective countermeasures.

3.2.1 History chart

The history chart is a specialized trend chart (§3.1.3), narrowed down to the latest four months of activity in the selected AS (i.e., until the last day of the current time range). Differently from the trend chart, the history chart includes a selector that allows to change

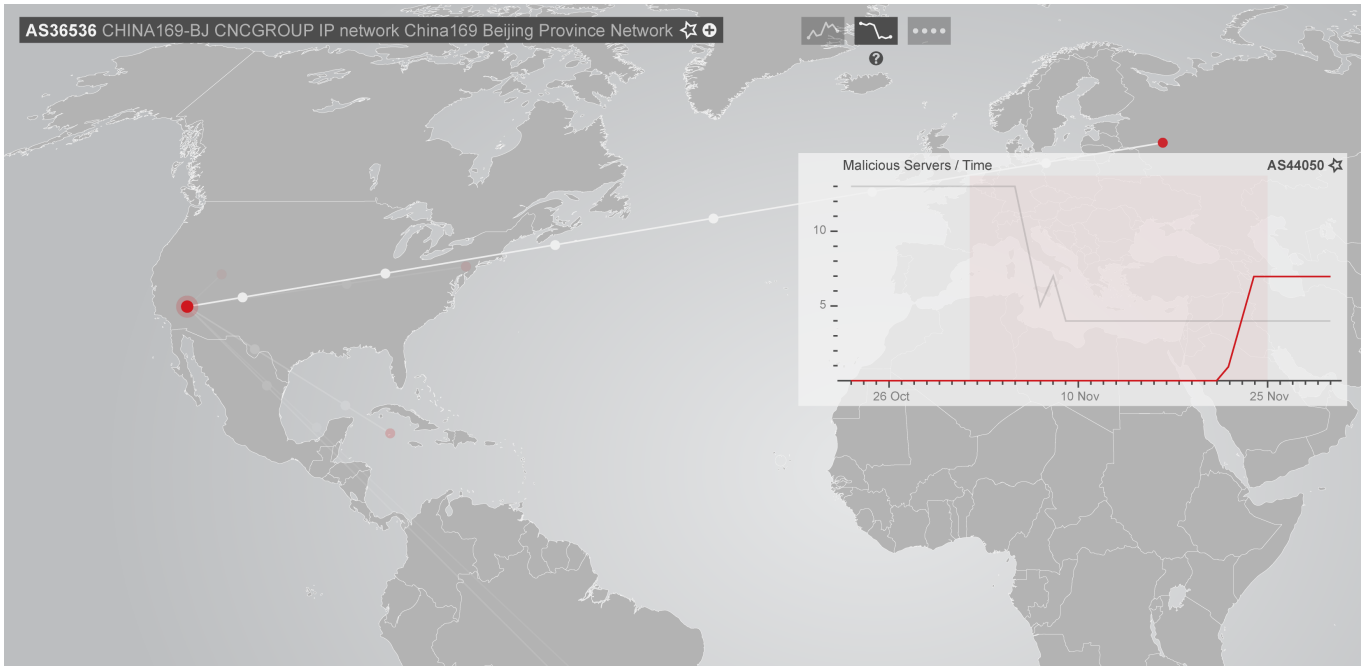


Figure 4: Cropped screenshot displaying a detected migration in the service migration screen. The frequency of the dots on the source-to-destination connecting line, and the thickness of the line itself are proportional to the confidence that the system has on the detected migration. The mini plot displays the number of hosts, which has dropped significantly in the source AS, whereas a sudden increase has occurred in the destination AS.

the plotted variable between number of servers in the AS, malicious score, and worldwide rank—with respect to the malicious score. The plot has daily granularity and, as the trend chart, one line is traced per type of activity, plus one for the overall sum.

3.2.2 *Service migration screen*

BURN provides a visual overview of the possible correlations between shutdowns, that are sudden drops of the number of hosts in the AS under examination, and activations, that are sudden increases of the number of hosts in other ASs.

Substantial drops in the daily number of hosts are listed as mini plots in the main dialog of the service migration screen. These *mini plots* are small trend charts sorted by the compatibility score, defined in §4.1.1, which represents the confidence that a migration has actually occurred. For example, an AS with a sudden drop of 100 IPs engaged in phishing activity has high compatibility with an AS that, in about the same time range, recorded a sudden increase of 100 IPs engaged in phishing activity. The geographical positions of the ASs involved in a migration are displayed on a geographical map when the user clicks on the corresponding mini plot.

As exemplified in Fig. 4, ASs are represented by non-animated dots. Lines connect the source of the migration, that is the AS that hosted the servers being shut down, to the destination ASs, where the malicious services have possibly migrated to. Animated impulses running over the connection lines represent the importance of the migration. More precisely, the frequency of the impulses and the line's opacity are both proportional to the compatibility score. Hence, connection lines characterized by high-frequency impulses draw the analyst's attention to relevant migrations.

The mini plot of the number of IPs over time in the source AS appears when hovering over the dot that represents said source AS. When hovering over the destination dot, a mini plot with both shutdown and activation in the same graph is displayed. The overlap of both the time series allows easy comparison of the events. At the bottom of the screen a box listing the other shutdowns occurred on the same AS allows a quick examination of other migrations.

3.2.3 *Service longevity chart*

Although the FIRE system pre-processes data sources to exclude ASs that show no persistent rogue activity, we found some ASs characterized by hosts with intermittent activity. This motivates the need for manual analysis, supported by BURN under the name of service longevity analysis. The core of this analysis is a *tolerance score* (defined in §4.1.2) that BURN uses to rank ASs, so that particularly "friendly" hosting providers are easy to spot as they will have high tolerance score. We complement this with an effective visualization of the activity of tolerant ASs, as described in the remainder of this section.

As exemplified by the screenshot in Fig. 5, the visualization for the longevity analysis consists in a timeline that depicts the daily activity of the malicious IP addresses of each AS: Rows correspond to IPs and columns correspond to days. In each cell, a red dot is displayed only when the IP is active. Dots may correspond to phishing, spam, malware, botnet activity, or any combination of these. Uninterrupted sequences of dots in a row highlight long-living hosts; if these are the majority, then the AS is taking no or ineffective countermeasures. In other words, the AS tolerates hosts that act maliciously.

4. SYSTEM DETAILS

Our system is built on top of the FIRE dataset, which contains synoptic data on rogue ASs as well as detailed data (e.g., IP, type of malicious activity) on the malicious hosts that contribute to the rogueness of each AS. For each AS, the number of malicious servers over time is known along with a maliciousness score—computed by the FIRE system as described in [7]—which summarizes the rogueness of each AS. The score is calculated on a daily basis is already

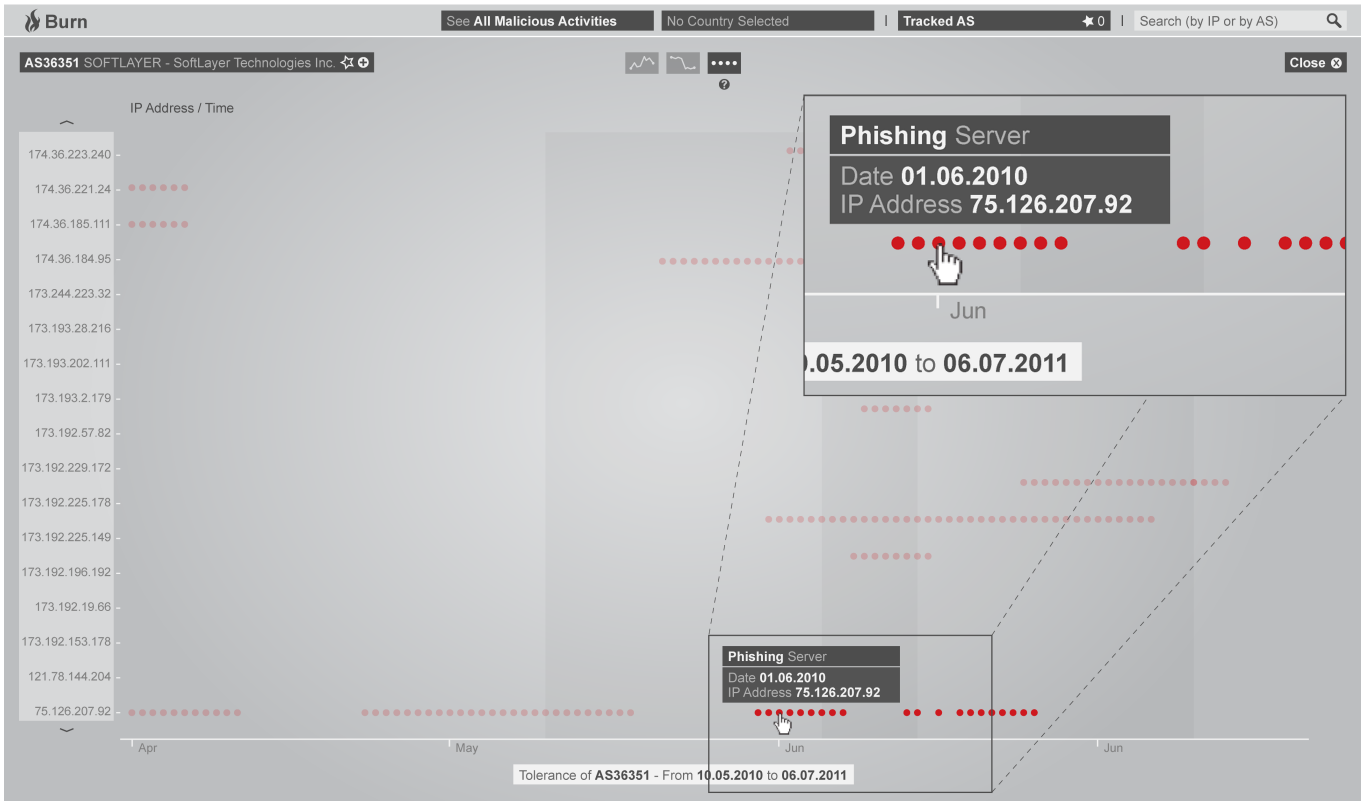


Figure 5: Screenshot displaying service longevity chart of a tolerant AS. Dots represent whether a given IP is active on each day, so that IPs characterized by persistent rogue activity are easy to spot.

normalized by the AS size.

BURN pre-processes and enriches the data from the FIRE database. Specifically, we calculate the integral of all numerical variables and keep them alongside the original, daily values, such that we can calculate time-based statistics (e.g., a monthly average) in constant time—necessary to pull the integral value of the first and last day of the given time range. We compute the integral of all the values online as new samples are streamed from the FIRE dataset. Also, we run daily the service migration and longevity analyses through a set of batch scripts, implemented over a shifting time window: To move ahead one day, it is enough to add the new data point and subtract the oldest one to each value, as detailed in §4.

We implemented the BURN back end upon a LAMP stack, whereas front-end computation runs on the client side, which implemented in Adobe Flash's ActionScript 3.0. The system details are described in the remainder of this section.

4.1 Rogue behavior analysis

4.1.1 Service migration

BURN uses a simple heuristic that recognizes signs of possible migrations between two ASs, from hereinafter called *source* and *destination* AS, respectively. Our key assumption is that migrations are composed by a *shutdown phase*, observable in the activity of the source AS, followed by an *activation phase*, observable in the activity of the destination AS. The shutdown phase happens either because the miscreants deactivate their servers in the source AS before transferring them to the destination AS, or because the source AS blocks the services—and thus the criminals migrate their services to the destination AS. When the number of malicious servers in some AS drops suddenly, followed by a corresponding sudden increase in the number of servers in another AS, BURN flags this pattern as a possible migration.

Each day i, called *current day*, we analyze the number of hosts in each AS through two sliding windows. First, we calculate the average number of malicious IPs in the AS within a window of size $\hat{W}$, called *observation window*, placed before the current day. This value is referred to as *observed average*, $\hat{\mu}$. Secondly, we calculate the same average within a window of size W, called the *current window*, placed right in between the observation window and the current day. We call this value the *current average*, μ. When the current average is significantly lower than the observed average, a shutdown has occurred. More precisely, we define the *current displacement* as $\delta = \hat{\mu} - \mu$, and threshold it with respect to the observed average. The key assumption is that shutdowns (1) involve a significant percentage of services and (2) take a relatively short time, i.e., $W < \hat{W}$, to complete. Therefore, we detect a shutdown whenever the *shutdown condition* $\frac{\delta}{\hat{\mu}} \geq \bar{\delta}$, is met, where $\bar{\delta} \in [0, 100\%]$ is the *minimum displacement* percentage, with respect to. For instance, if $\bar{\delta} = 70\%$, and $\hat{\mu} = 1{,}000$, then a shutdown is detected when the current average $\mu \leq 300$. Both the average values and the displacement are functions of the current day i and AS a under examination. More formally, $\mu = \hat{\mu}_i = \hat{\mu}_i(a)$, $\mu = \mu_i = \mu_i(a)$, and $\delta = \delta_i = \delta_i(a)$, where $a \in \mathbb{AS}$ denotes an AS, and $\mathcal{AS}$ is the set of all the ASs. Fig. 6 summarizes variables and parameters hereby defined. We define the *shutdown set* $\mathcal{S}_i$ as the set of those ASs that satisfy the shutdown condition:

$$\mathcal{S}_i = \left\{ a \in \mathcal{AS} \mid \frac{\delta_i(a)}{\hat{\mu}} \geq \bar{\delta} \right\}$$

These ASs are candidate sources for possible migrations.

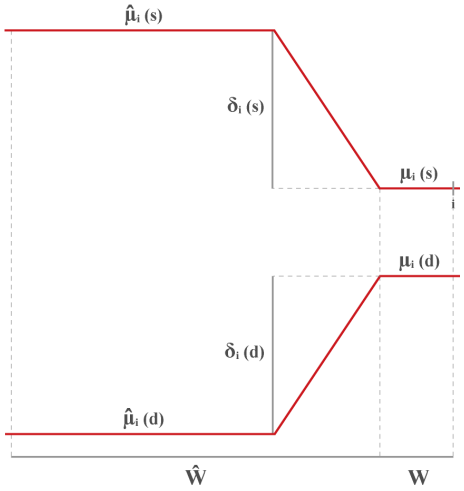


Figure 6: Observation window $\hat{W}$ and average $\hat{\mu}$, current window W and average μ, and displacements δ_i, in an hypothetical shutdown s (top) and corresponding activation d (bottom).

We find migrations by searching, for each ASs in shutdown phase, another AS that has the same displacement, but opposite in sign (i.e., activation phase). In ideal conditions, this means that, for a given source AS s, we search a destination d such that $\delta_i(s) = -\delta_i(d)$. More precisely, we define the *migration set*, as follows:

$$\mathcal{M}_i = \left\{ (s,d) \in \mathcal{S}_i \times \mathcal{AS} \backslash \{s\} \mid \begin{array}{l} \delta_i(s) > 0 \\ \delta_i(d) < 0 \end{array} \wedge \frac{|\delta_i(s) + \delta_i(d)|}{\delta_i(s)} \leq \bar{\Delta} \right\},$$

where $\delta_i(s)$ and $\delta_i(d)$ are the displacement of the (candidate) source AS and destination ASs, respectively, and where $\bar{\Delta} \in [0, 100\%]$ is a threshold to account for small variations from ideal conditions. Specifically, if $\bar{\Delta} = 0\%$ then $\mathcal{M}_i$ would contain only source-destination couples such that $\delta_i(s) = -\delta_i(d)$. In other words, this threshold accounts for the fact that the miscreants may transfer only part of the services, neglecting some, and we wish to allow for some flexibility.

Compatibility score.

BURN ranks the migrations by means of a compatibility score, to prioritize the manual review process described in §3.2.2. We first define the *compatibility* as an ordering relation "$<_C$" defined on a migration set $\mathcal{M}_i$ by means of the compatibility function $C^{(j)} : \mathcal{S}_i \times \mathcal{AS} \mapsto [0,1]$ between source and destination ASs, with respect to malicious activity of type $j \in \mathcal{J} = \{\text{phishing, malware, spam, bot}\}$. In simple words, the compatibility function quantifies the discrepancy between two ASs in terms of each type of activity. For instance, an AS with 10 phishing services and 5 malware services being shut down is more compatible with an AS having 10 new phishing services and 5 new malware services, than with an AS having 9 new phishing services, 5 new malware services and 3 spam services being shut down. More formally

$$C^{(j)}(s,d) := \frac{\min_{a \in \{s,d\}} \delta^{(j)}(a)}{\max_{a \in \{s,d\}} \delta^{(j)}(a)},$$

where $\delta^{(j)}_{min}$ and $\delta^{(j)}_{max}$ are the minimum and the maximum values of $\delta^{(j)}(\cdot)$. This function quantifies the ratio between the minimum and the maximum displacement, in number of services, of type j. Note that $\delta^{(j)}(\cdot)$ denotes the displacement calculated only for the activity of type j. At this point, we define the overall *compatibility score* as the weighted average of the compatibility function calculated over all the possible activities in $\mathcal{J}$, that is

$$C_{s,d} := \frac{\sum_{j \in \mathcal{J}} C^{(j)}(s,d) \cdot \delta^{(j)}(s)}{\sum_{j \in \mathcal{J}} \delta^{(j)}(s)}$$

where the weights are the displacements in the source AS.

4.1.2 Tolerance to long-living rogue hosts

Some ASs host servers that exhibit persistent malicious activity for long periods of time, as opposed to more controlled ASs, which are instead characterized by short-living rogue servers. We compute the service longevity as the number of days that a server remains active; this number is then normalized by τ (FIRE requires that a server remains active at least τ days to deem its AS potentially rogue). More formally, for a given AS $a \in \mathcal{AS}$ we define the *tolerance* with respect to a host $IP \in a$ as the longevity of the IP:

$$T_i(IP) := \frac{\sum_{t=0}^{\tau} \mathbf{1}_{IP}(i+t)}{\tau}$$

where the indicator function $\mathbf{1}_{IP}(i+t)$ is one only when IP is active, and i is the current day. We calculate *tolerance* of the entire AS as:

$$T_i(a) := \frac{\sum_{IP \in a} T_i(IP)}{|a|}$$

where $|a|$ denotes the number of malicious servers in the AS a. This value is calculated by the FIRE system.

4.2 Visualization techniques

Before detailing the visualization techniques adopted in BURN, it is important to underline key choices that influenced its design. Our main goal, in terms of visualization, is to overcome the limits of the variable-encoding model, based on traditional visual variables (e.g., color, shape, size), which are not as effective as visual structure, physical layout and visual dynamics are in easing pattern recognition, grasping the overall structure of the phenomenon, and finally fostering insights [9]. Thus, we decided to reduce the importance of color and concentrate on the visual interaction with the timeline, the visual layout of all the elements, and their animation patterns. Unlike many security visualizations systems, which use different colors to represent different degrees of security or insecurity (e.g., risk or threat level), we purposefully used a single color (i.e., red). The reason is twofold. First, colors near the red spectrum are naturally connected to risk feelings, hence suitable for describing threat scenarios. Second, different colors may mislead users, leading them to link different colors to different levels of maliciousness or importance. Instead, we use different color shades, as discussed in §3.1.2, and simple animations §4.2.5 to visualize different degrees of insecurity. Last but not least, vision deficiencies and other disabilities such as color blindness are mitigated by using only one color.

4.2.1 Filters

Highlight filter.

The bubble chart provides a customizable mechanism to highlight extra variables other than the number of malicious hosts: service shutdowns, and service longevity. The filter acts by hiding those ASs not meeting the selected variable, by decreasing their opacity. This mechanism is called highlight filter and is useful to elicit phenomena of interest for the analysis, without leaving the focus from the main variable projected on the vertical axis. The highlight filter is available at countrywide zoom, and allows the user to select events to stress.

Figure 7: Activity and country filters. When clicked, these allow to narrow the current visualization to the type of data selected.

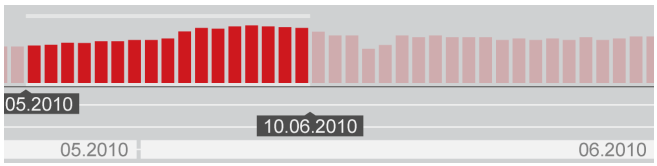


Figure 8: The timeline and range selectors allow to narrow the visualization to a given period of time. These controls are omnipresent and hide automatically when not needed.

The combination of custom variable selection (described in §3.1) and highlight filters allows the analyst to rearrange the bubble chart and the geographical map dynamically with respect to several aspects, while keeping under control the "driving" direction of the analysis.

Activity filter.
The activity filter allows to select, from a drop-down menu, the type of malicious activity visualized: "C&C" refers to command and control traffic performed by bots, "Malware" regards IPs engaged in hosting drive-by downloads exploits and malware, while "Phishing" and "Spam" refer to IPs engaged in phishing and spamming, respectively. By default, the overall aggregation of all malicious activities is displayed.

When (de)activating an activity, the global view visualizations are updated instantly. Notably, comparing the selected activity with the overall aggregation is still possible, since the overall aggregation is shown on the background of the bubble chart, country level map, and timeline. This is useful to highlight the contribution of each selected activity on the overall malicious activities.

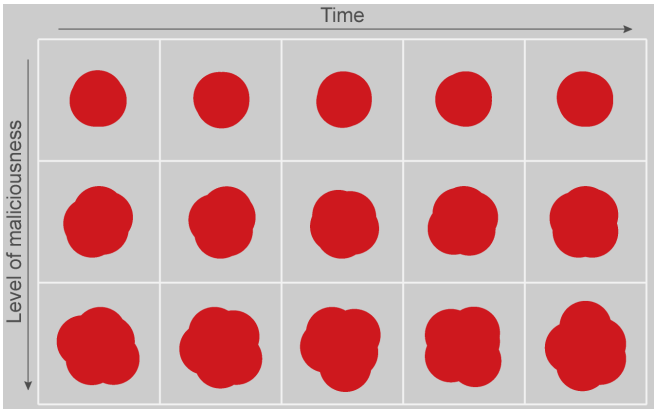
4.2.2 *Timeline and time range selection*

The latest week of activity in the database is displayed by default in the global view screens. The timeline, depicted in part in Fig. 8, is at the bottom of both the bubble chart and the geographical maps for quick time range selection. The selection of the time range, from one day, to several months, can be carried out by positioning two sliders indicating the range's start- and end-point. The selection impacts automatically all the visualizations present in the system, ensuring coherent global observation of phenomena.

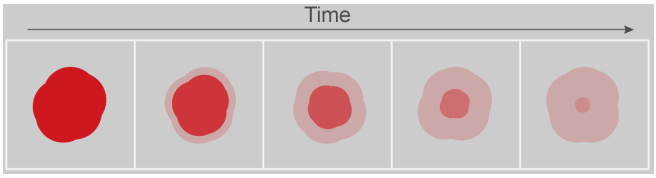
The timeline is a bar chart that summarizes the daily number of malicious hosts of the selected activity type. If no activity type or country are selected, each bar represents the total number of malicious hosts worldwide.

4.2.3 *Contextual dialogs*

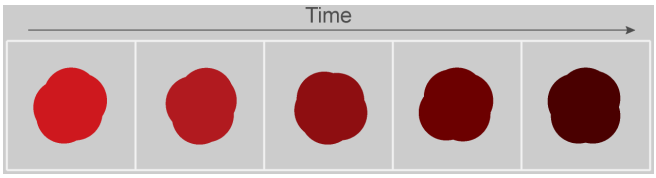
When clicking on the bubble of an AS from the bubble chart (e.g., Fig. 1) or geographical map, a contextual modal dialog displays summary information about the AS, such as its average malicious score, size, geographic location, number of malicious IPs, and a star-shaped button for adding the AS to the tracking list. This dialog can be expanded to look at more information: the AS name, registered owner, and a breakdown of malicious activity types (e.g., 30% phishing, 20% spam, 50% botnet traffic). In addition to the contextual dialog, which appears at the left of the AS bubble, three buttons on the right-hand side allow quick access to the history chart, service migration screen, and longevity analysis chart. The size of the AS (i.e., number of /20 IPs) is represented by a background circle, of proportional area, centered around the bubble. This allows direct comparison of number of malicious machines vs. maximum number of IPs the AS can host.



(a) Increasing maliciousness.



(b) Shutdown events.



(c) Increasing tolerance.

Figure 9: Animation effects used to represent an AS's (a) maliciousness, (b) shutdowns, and (c) tolerance.

4.2.4 *Autonomous system tracking list*

The user can mark ASs for further inspection without deviating the focus of the analysis, by adding them to the global autonomous system tracking list. This is performed by clicking on the star-shaped button that appears on each AS's contextual dialog. As a result, tracked ASs are marked with a star symbol. All tracked ASs are also listed in an omnipresent drop-down menu. This feature is particularly useful to track relevant ASs globally, because it creates a lightweight bridge across different visualizations that implements one of the underpinning goals behind BURN, that is to provide a continuous switch between different data displays.

4.2.5 *Animations*

Animations are adopted in BURN for three purposes. First, for enriching the bubble chart with extra variables without inserting further, possibly confusing graphical elements (e.g., changing the size of the bubble, or using multiple colors). Secondly, animations turn out to be effective for conveying concepts such as ASs' activity

intensity, presence of shutdowns, and long-living malicious hosts. Third, animations are used for drawing the attention of the user to important events. As the behaviors are not mutually exclusive, the following animations can occur simultaneously on the same bubble leading to non-confusing visualizations. More precisely, the first animation is always active, whereas the last two act as visual filters that highlight the corresponding behaviors.

Maliciousness For monitoring the malicious score of each AS and, at the same time, convey the idea of maliciousness as related to the magnitude of activity, each AS bubble is formed by a set of overlapping circles that continuously shift their position, both horizontally and vertically, with offsets proportional to the AS's maliciousness score. In other words, the wider bubble "vibrates", the more the AS is rogue. As shown Fig. 9a, this effect conveys the perception of ASs as entities composed by different, active and interacting elements (i.e., malicious hosts). The animation is active in bubble chart and country-level geographical map, and is especially useful when the bubbles are ordered by other variables (e.g., increments or AS size), since in these situations the malicious score is not indicated by the bubbles' vertical position.

Shutdown ASs in shutdown phase in the current time range are displayed with bubbles of progressively-decreasing size. This effect loops continuously so that ASs that exhibit shutdowns phase are easily recognizable at a first glance. The original size of the bubble is always visible, also during the animation, in background.

Tolerance The animation changes the color brightness of the bubble of those ASs that are characterized by tolerant behaviors within the time range selected. The brightness decreases gradually until the bubble is almost black, then the animation starts again, in a continuous loop.

5. USE-CASE SCENARIOS

In this section we describe three typical scenarios supported by BURN fed with *real* data from FIRE. More precisely, we used the portion of database populated by FIRE between January and December 2010—this data, including AS numbers and host IPs, is publicly available via `http://maliciousnetworks.org`. In this time range, our system recognized 240 shutdowns in one year, leading to about 11,000 possible migrations. The main reason for this large amount of migrations is that we purposely set a small $\bar{\delta}$, such that, small migrations (e.g., 4–5 services) are taken into account. However, the compatibility score allows to rank relevant migrations, which are typically 5 to 10 per shutdown. Unfortunately, there is no ground truth for validating migrations, otherwise there would be no motivation behind our migration detection algorithm.

5.1 Finding topmost malicious ASs

The analyst needs to examine highly malicious ASs in 2010. To this end, he or she first finds the topmost malicious AS overall, and the topmost malicious AS per activity type.

When started, the application displays the last week of activity in the bubble chart (e.g., Fig. 1) by default. To find peaks of malicious activity in the whole year, the user inspects the trend chart (Fig. 3), which displays the aggregated degree of malicious activity over time. At this scale, the peak of activity between the end of May and the first half of June is easy to recognize visually. By leveraging the interactive timeline at the bottom of the bubble chart, such time range is quickly selected and the bubble chart is updated instantly, displaying the topmost malicious AS, in said time range, at the top of the ranking.

These few clicks quickly lead to AS21844, which is the topmost malicious AS in the time range around the peak of activity. It is interesting to notice that larger yet less malicious ASs exist in the ranking. Summary information about AS21844 is displayed by clicking on the corresponding bubble, triggering the contextual dialog. Additionally, the second-, third-, and fourth-most malicious ASs are starred to the tracking list. When finished analyzing AS21844, the user can rely on the omnipresent list of tracked ASs to quickly skim through the other suspicious ASs and lead to the interesting finding that the third-most malicious AS overall, AS21740, has been hosting machines engaged in substantial botnet traffic. To confirm this, the user switches back to the bubble chart and, by using the activity filter (Fig. 7), focuses on C&C activity only, for which AS21740 is updated at the topmost position. This corroborates the conjecture that AS21740 is the global leader in C&C activity within the peak time range. By switching on and off the filters for other types of activity (e.g, phishing, malware, and spam) the user can quickly visualize sliced rankings and analyze the ASs from other perspectives.

5.2 Tracing attacks

This scenario underlines how BURN makes it easy for the user to investigate the context in which an isolated event occurred. Specifically, a suspicious email containing a possibly phishing URL draws the security expert's attention, because the corresponding IP address (75.126.207.92, resolved with regular whois tools) turns out to point to a server connected to an AS considered tolerant, letting such server run continuously for several days.

The analyst visualizes the last week of data in the worldwide geographical map (Fig. 2) to overview the global status and obtain other specific information by hovering on some countries—thus triggering corresponding contextual dialogs. The search box, on the top-right corner of the screen, is leveraged to pinpoint the suspicious IP's AS. When found, the system automatically zooms in over the Germany area (bottom right of Fig. 2), where said AS's contextual dialog is opened to highlight the search result and show detailed information.

At the same time, because of the maliciousness animation (Fig. 9a), the analyst notices the AS among the others on the country map. The expert's attention is also drawn by the AS bubble's color changing animation, which decreases its brightness gradually. A label on the right-hand side of the bubble suggests to check the AS's tolerance by opening up the longevity chart, where the IPs of the AS can be browsed by simple vertical scrolling, until the IP of interest is found. In this chart, displayed in Fig. 5, long-living hosts are characterized by extended sequences of adjacent dots that, when hovered, display a tooltip with the IPs' top activity on each day. As suspected, the server in question has been exhibiting phishing activity recently, while previously it was serving as a bot, noticed when hovering on one dot, as the opacity of dots not matching the same type of top activity decreases.

5.3 Finding and highlighting migrations

In this scenario, a security officer monitors the activity of a botnet engaged in several DDoS attacks. The botnet is known to be controlled by several bot masters located in AS36536's network. These servers, however, have been recently reported to be inactive. The goal of the security expert is to assess the correctness of such reports.

Differently from the previous scenario, where the search box was used from the geographical map—thus triggering an automatic zoom in at country level—the AS is new looked up via search box from the bubble chart. This activates the AS's contextual dialog on the bubble. The animation exemplified in Fig. 9b on the bubble indicates the presence of substantial shutdowns in progress. The user clicks to the

contextual dialog's shutdown icon, switching to a list of mini-plots displaying the time intervals during which shutdowns have likely occurred.

The user clicks on the first (i.e., most recent) shutdown. This opens the service migration screen, where a geographical map shows the location of the source AS and corresponding possible destinations (Fig. 4 shows part of the service migration screen used in this scenario). When hovering on the red dot of the source AS, the mini-plot detailing the shutdown appears. At this point, manual review of the possible migration alternatives is as easy as moving the pointer over the destination dots. As shown in Fig. 4, the shutdown and activation mini plots are overlapped and displayed in translucency. Two of the connection links are characterized by particularly frequent impulses an a high-opacity destination dot. This visual pattern suggests significant source-to-destination compatibility (e.g., close to one) and indeed draws the analyst's attention. As the connection lines and the destination dots look very similar to each other, the migration under examination might involve multiple (i.e., two) destinations: AS44050 and AS32592.

6. LIMITATIONS

Although the main purpose of the bubble chart (described in §3.1.1) is to provide a ranked overview of ASs, it may become cluttered when then number of ASs with similar scores is high. In addition, the differences between dissimilar, yet close, malicious scores can be hardly depicted. In the current implementation of BURN we left this issue behind, and focused on other, more substantial visualization-related problems. In fact, in about one year of collected data, this problem never arose. Should this issue become significant, it could be addressed by including a "stress feature", i.e. a way to magnify such differences, and by applying a non-linear adaptive scale aimed at optimizing the visualization of actual data ranges. In this way, the distribution of the ASs on the bubble chart could highlight even those differences that are difficult to detect.

Secondly, the migration detection heuristic does not guarantee that what it finds are actual service migrations. However, the goal of this function is to recognize signs that may indicate migrations, which are meant to be reviewed manually. To reduce the effort required for this inspection, migrations are ranked by a compatibility score, to help the security expert to prioritize the review process, focusing on more plausible migrations first. This limitation could be further alleviated by incorporating more data sources, such as reports of attacks captured by honeypots (e.g., CaptureHPC) and malware analyzed in sandboxes (e.g., Anubis, Wepawet), and by correlating these data for confirming whether or not a migration has occurred. For instance, by observing the network connections established by a bot sample running in a sandbox, one may find that the targets of the connections (e.g., the bot-master) changed suddenly.

7. CONCLUSIONS

Considering visualization as part of a communication and understanding process, BURN has been designed to let the user move easily between multiple views, without cluttering the screen or dictating specific data analysis paths. Particular attention has been paid to the rhetorical potential of visualization. Specifically, lightweight animations have been incorporated to create intuitive and immediate mappings, allowing the simultaneous representation of up to four variables. The system also includes a novel migrations detection algorithm that helps researchers in root-cause investigations.

Three use-case scenarios built and tested on real world data have been presented, to better explain the features and the interaction modes provided by our system, based on some of the most common experts' needs. From these scenarios, which represent just an excerpt of the full capabilities of BURN, it emerges that our system is both effective at aiding the user to pursue his or her goals and extremely flexible. Basically, BURN imposes no methodology to expert analysts, while end-users are provided with an intuitive interface, requiring no prior knowledge to spot relevant events.

Future efforts will focus on refining the migration detection heuristic to reduce the amount of candidate migrations, thus minimizing manual inspection efforts. To this end, we plan to correlate found migrations with other data sources (e.g., attacks captured by honeypot) to confirm detected migrations. In addition, we are designing a usability study experiment, mostly based on the use-case scenarios presented, which we will conduct on our prototype.

Acknowledgments

The authors are thankful to the FIRE developers for their support. This work has been partially supported by the European Commission through IST-216026-WOMBAT funded by the 7th FP. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the European Commission.

8. REFERENCES

[1] L. Harrison, X. Hu, X. Ying, A. Lu, W. Wang, and X. Wu. Interactive detection of network anomalies via coordinated multiple views. In *Proc. of the Seventh Intl. Symposium on Visualization for Cyber Security*, VizSec '10, pages 91–101, New York, NY, USA, 2010. ACM.

[2] D. Inoue, K. Yoshioka, M. Eto, M. Yamagata, E. Nishino, J. Takeuchi, K. Ohkouchi, and K. Nakao. An incident analysis system nicter and its analysis engines based on data mining techniques. In *Proc. of the 15th intl. conference on Advances in neuro-information processing - Volume Part I*, ICONIP'08, pages 579–586, Berlin, Heidelberg, 2009. Springer-Verlag.

[3] G. Kontaxis, I. Polakis, S. Antonatos, and E. P. Markatos. Experiences and observations from the noah infrastructure. Technical report, ICS, FORTH, 2010.

[4] R. Marty. *Applied Security Visualization*. Addison-Wesley Professional, 1 edition, 2008.

[5] L. Masud, F. Valsecchi, P. Ciuccarelli, D. Ricci, and G. Caviglia. From data to knowledge - visualizations as transformation processes within the data-information-knowledge continuum. *Information Visualisation, Intl. Conference on*, 0:445–449, 2010.

[6] A. H. Robinson, J. L. Morrison, P. C. Muehrcke, A. J. Kimerling, and S. C. Guptill. *Elements of Cartography*. Wiley, 6 edition, Mar. 1995.

[7] B. Stone-Gross, C. Kruegel, K. Almeroth, A. Moser, and E. Kirda. Fire: Finding rogue networks. In *Proc. of the 2009 Annual Comp. Security Applications Conference*, ACSAC '09, pages 231–240, Washington, DC, USA, 2009. IEEE Comp. Society.

[8] T. Yu, R. Lippmann, J. Riordan, and S. Boyer. Ember: a global perspective on extreme malicious behavior. In *Proc. of the Seventh Intl. Symposium on Visualization for Cyber Security*, VizSec '10, pages 1–12, New York, NY, USA, 2010. ACM.

[9] C. Ziemkiewicz and R. Kosara. Beyond bertin: Seeing the forest despite the trees. *IEEE Comp. Graphics and Applications*, 30:7–11, 2010.