Web Defacement Campaigns Uncovered

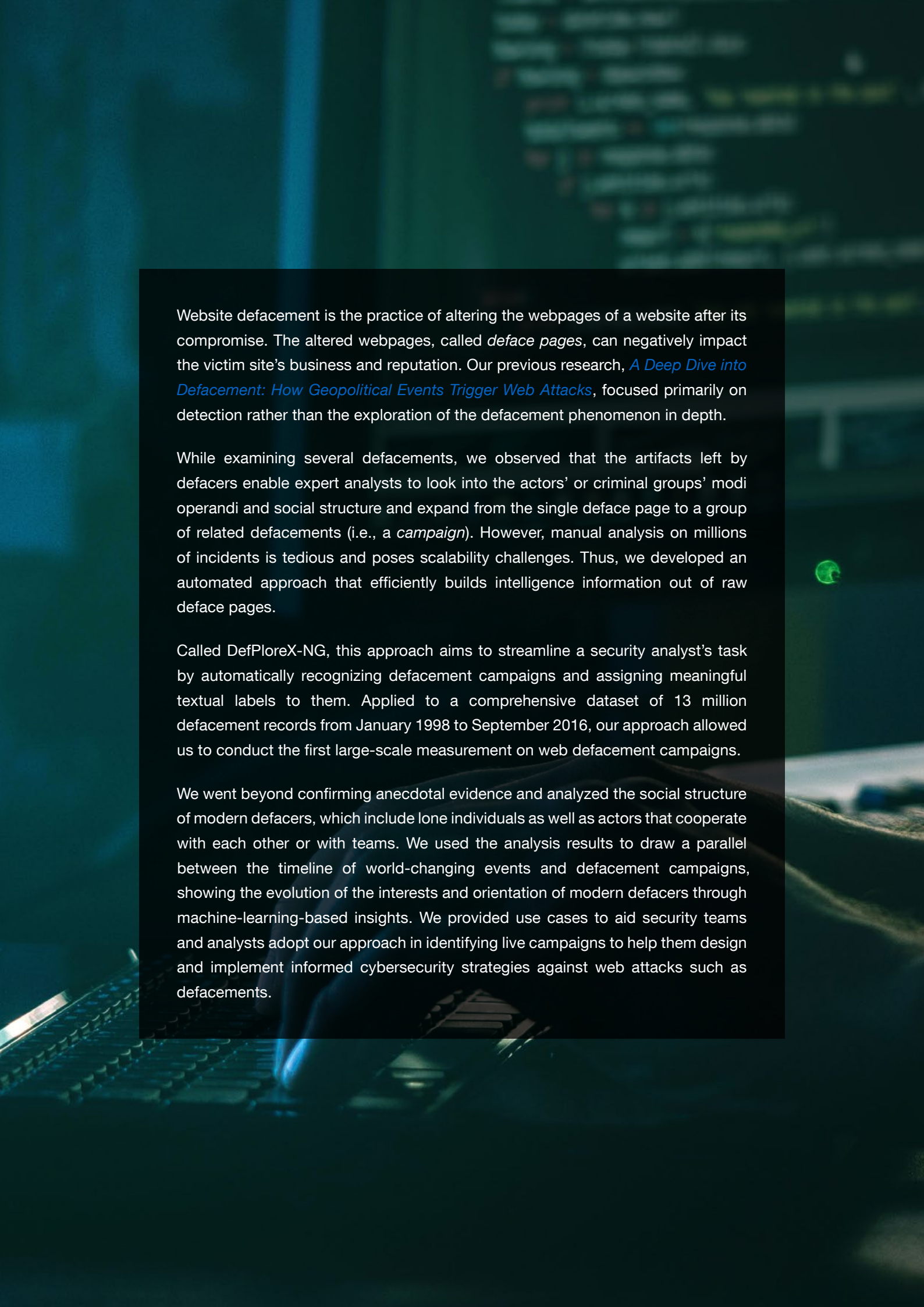
Gaining Insights From Deface Pages Using DefPloreX-NG

Federico Maggi, Marco Balduzzi, Ryan Flores, Lion Gu, Vincenzo Ciancaglini
Trend Micro Forward-looking Threat Research (FTR) Team

A **TrendLabs**SM Research Paper

for Raimund Genes (1963-2017)

Contents



Website defacement is the practice of altering the webpages of a website after its compromise. The altered webpages, called *deface pages*, can negatively impact the victim site's business and reputation. Our previous research, *A Deep Dive into Defacement: How Geopolitical Events Trigger Web Attacks*, focused primarily on detection rather than the exploration of the defacement phenomenon in depth.

While examining several defacements, we observed that the artifacts left by defacers enable expert analysts to look into the actors' or criminal groups' modi operandi and social structure and expand from the single deface page to a group of related defacements (i.e., a *campaign*). However, manual analysis on millions of incidents is tedious and poses scalability challenges. Thus, we developed an automated approach that efficiently builds intelligence information out of raw deface pages.

Called DefPloreX-NG, this approach aims to streamline a security analyst's task by automatically recognizing defacement campaigns and assigning meaningful textual labels to them. Applied to a comprehensive dataset of 13 million defacement records from January 1998 to September 2016, our approach allowed us to conduct the first large-scale measurement on web defacement campaigns.

We went beyond confirming anecdotal evidence and analyzed the social structure of modern defacers, which include lone individuals as well as actors that cooperate with each other or with teams. We used the analysis results to draw a parallel between the timeline of world-changing events and defacement campaigns, showing the evolution of the interests and orientation of modern defacers through machine-learning-based insights. We provided use cases to aid security teams and analysts adopt our approach in identifying live campaigns to help them design and implement informed cybersecurity strategies against web attacks such as defacements.

Overview

Web defacement, or simply *defacement*, is the practice of visibly altering one or more webpages of a website upon compromising it. The intent of the actor, called *defacer* in this context, is to arbitrarily change or replace the original content of the victim website to advertise the success of the compromise. The resulting page, called *deface page*, may contain the following information: motive behind the attack, team affiliation of the defacer/s, and/or aliases of the supporting actors. Over the years, defacers have abandoned their interest in defacing webpages for the sheer purpose of advertising the success of their compromise, now pursuing defacement more as a means to broadcast strong messages to the world — by compromising popular websites.

While several actors are still driven by the desire to deface webpages to promote their reputation, an increasing number of defacers have begun to use defacement to promote their ideologies, religious orientation, political views, or other forms of activism, often following recent real-world events (for example, wars, elections, crises, terrorist attacks). We refer to this phenomenon as *dark* propaganda to highlight the fact that legitimate resources are being abused to promote the actors' viewpoints. For example, from 2013 to 2014, Team System Dz defaced over 2,800 websites and planted pro-ISIL/ISIS (Islamic State of Iraq and the Levant / and Syria) content to protest U.S. military involvement in the Syrian Civil War.[1] In one of these incidents, the actors altered[2] the homepage of Keighley Cougars, a British Rugby League club team, to display disturbing war-related pictures, along with an "I love you ISIS" text. Another incident addressing a different issue took place in January 2014, when news broke about the U.S. and British intelligence agencies' alleged collection of data from mobile applications, including Angry Birds.[3] A group of defacers vandalized the game's homepage with a "Spying Birds" text and the NSA logo.

The inappropriate or offensive content placed by defacers affect the reputation and continuity of the legitimate businesses behind the targeted websites, making these campaigns not as innocuous as they appear. In response to this, past research works have proposed monitoring solutions[4] [5] [6] [7] to *detect* defacement content whenever it is being planted on a website. Although these systems are operationally useful, they do not provide in-depth knowledge to help analysts *understand* the web defacement phenomenon from an investigative standpoint. For example, one detection method[7] uses the most accurate yet very *opaque* machine-learning technique — convolutional neural networks (CNNs)[8] trained via deep learning. Thus, despite yielding high detection rates, the outcome is not very helpful to analysts who need to *track down* the actors behind web defacement attacks or reconstruct a campaign of defacements.

We have observed a lack of methodologies that can analyze, understand, and track web defacements at large. Previous research along this line relied on metadata alone (for example, information pertaining to attacker intention, vulnerability used, and alias of the attacker). Such metadata is spontaneously provided by the actor and should not be considered trustworthy. The only researches that have inspected the *content* of the deface page are either outdated (a study[9] was made in 2004, but web defacement has evolved since then) or limited to manual inspection of a handful of pages.[10] Clearly, there is a need for a comprehensive and large-scale analysis approach, especially given the availability of datasets spanning over 19 years. So far, these datasets have only been used for detection purposes.

To fill this gap, we chose to take a data-driven approach that supports analysts in exploring and visualizing the existing data, eliciting relevant web defacement campaigns. In this approach, we identify a set of core characteristics that translate the unstructured content planted by defacers (that is, webpage and linked resources) into useful features. Using such features to represent a large dataset, we then feed a data-analysis and machine-learning pipeline that maximizes scalable clustering techniques to automatically group related defacement records into campaigns. Next, we label the identified campaigns in a convenient and human-readable form so that analysts can easily use the results for classic data exploration tasks (e.g., graphing, plotting, drilling, and pivoting).

The overall system is called DefPloreX-NG (a play on the phrase "defacement explorer," with the acronym for Next Generation appended to it to signify advancement). As a system and toolkit for identifying and tracking web defacement campaigns in historical and live data, it not only provides analyses that lead to the detection of relevant campaigns but also sheds light on the actors' modi operandi, social structure, and organization. In addition, defacements can be tracked and studied with flexibility. Operationally, analysts and other security professionals can use the results to provide early warning (for example, to sites that are more prone to defacement), to understand thoroughly how defacement happens (given knowledge of upcoming geopolitical events), and to conduct an ex-post study.

Dataset of Defacement Records

This paper used a dataset comprising a unique collection of defacement records from five major reporting sites (summarized in Table 1). These reporting sites provide feeds of defacement records, aggregated from various sources, such as sharing initiatives, Computer Emergency Readiness Teams (CERTs)/ Computer Security Incident Response Teams (CSIRTs), or victims. Often, defacers themselves voluntarily submit their defacements to advertise and show off their mischief. Defacers are generally interested in submitting correct data, to the extent that certain web exploit kits include submission routines that notify the reporting sites automatically upon successful execution of the payload.[11] However, while maintainers of popular sites strive to manually validate each defacement report, there is no assurance that the data will be free from deliberate or incidental errors.

Source	Site URL	Number of Records
Zone-H	www.zone-h.org	12,303,240
Hack-CN	www.hack-cn.com	386,705
Mirror Zone	www.mirror.zone.org	195,398
Hack Mirror	www.hack-mirror.com	68,980
MyDeface	www.mydeface.com	37,843
Total		12,992,166

Table 1. Number of records per reporting site

As shown in Table 2, each defacement record consists of metadata (e.g., timestamp of the defacement event, target URL) and raw content (e.g., the planted HTML or multimedia content). To be able to conduct our analysis, we also derived additional attributes such as the category of the defaced site (e.g., news, media), the top-level domain (TLD), and so on.

Type	Attribute	Example	Description	Trustworthiness	Explanation
Metadata (~1GB)	Timestamp	1998-01-02T15:14:12+00:00	Time of reported defacement incident	Medium–high	The attacker will get poor visibility by forging this datum
	Nickname	Team CodeZero	Pseudonym of the attacker or submitter	Low–medium	The attacker has no interest in, but can forge, this datum
	URL	http://janet-jackson.com/	URL of the planted deface page	High	The attacker has no interest in forging this datum and can be verified by the submission site
	Web server	Nginx	Name and brand of the web server running at the time of the defacement	Low	The attacker or submitter can forge it at no cost
	Reason	Political reasons	Motive of the defacement	Low	The attacker or submitter can forge it at no cost
	Hack mode	SQL injection	Vulnerability used to enable the upload of the defacement content	Low	The attacker or submitter can forge it at no cost
Raw content (~1TB)	Main page	HTML or TXT file	File storing the source code of the main defaced content (at the given URL)	High	The planted content is the subject of the defacement
	Embedded resources	Various formats	Images or other web resources linked by the defaced page and hosted within the same compromised web server	High	Collected as part of the main page by the submission site
	External resources	Various formats	External resources used in the main page	Medium–high	Can change over time or become unavailable

Table 2. Metadata and raw content in the dataset, along with a description of the trustworthiness of each attribute

Zone-H is the largest and richest archive because of its popularity and because it receives cross-submitted contributions. Thus, we purchased a copy of its dataset (up to September 2016), along with screenshots of deface pages captured at the time they were reported. The entire dataset takes about 1GB for metadata in CSV format while the screenshots of deface pages are almost 1TB. We complemented Zone-H's dataset by also crawling the other reporting sites listed in Table 1.

As shown in Figure 1, our collection of records spans over almost 19 years (January 1998 to September 2016). In this timeframe, the number of reported incidents per year grew from thousands to more than a million.

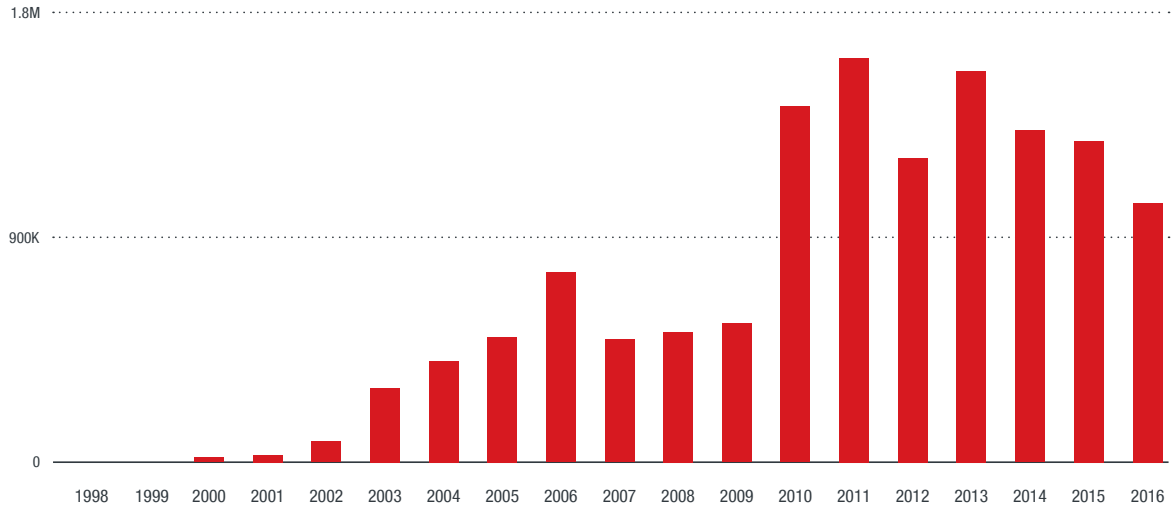


Figure 1. Records per year from Jan 1998 to Sep 2016

Metadata versus Content

Given the heterogeneous and possibly unknown origins of the data from Zone-H, these feeds are to be used cautiously, especially in operational environments, because the risk of false or misleading information is quite high. As highlighted in Table 2, there are various degrees of trustworthiness for each data attribute. Although we used metadata attributes in this research to draw out statistics or trends, the core of our analysis is based exclusively on the actual content planted by the attacker on the deface pages (e.g., HTML text, images, URLs, other linked resources, both internal and external). We consider the actual content as the most reliable data.

A set of deface pages set up by a group of actors — with a given goal in mind — is referred to as a campaign. The concepts required to better define the meaning of campaign are discussed in the next section.

For this research, all metadata with low trustworthiness, such as the defacer's declared nickname, target web server, exploited vulnerability or hack mode, and reason for hacking, are ignored. This is one of the core differences between our approach and previous work (see Reference [10]). The timestamp, which the attacker *cannot* forge and therefore has a medium to high trustworthiness, is ignored. The attacker might still wait for a certain amount of time before submitting a defacement report, but that would be against the attacker's interest. Moreover, we do not fully trust the accuracy of the timestamp nor are we using it as a feature.

Overall Statistics and Trends

This section provides an overview of the key statistics and trends that we observed in our dataset before the automated analysis was performed.

Topics Over the Years

To observe how the messages left by defacers evolved over the years, we used an off-the-shelf machine learning technique called *topic modeling*, which is widely used in news classification to determine the subject of a written story (e.g., politics, technology, finance).

For the scope of this section, a topic modeling algorithm can fit a large quantity of documents (i.e., our deface pages) to an arbitrarily small set of high-level concepts. We used the latent semantic analysis technique,[12] which assumes that words that are close in meaning will also appear in similar documents.

In practice, after extracting the text from each of the planted webpages, we removed stop words (taken from a comprehensive list at https://github.com/igorbrigadir/stopwords), words containing non-ASCII letters, and any occurrence of team or defacer nicknames (from the metadata) as these don't contribute to defining the overall message. Subsequently, we fed the topic modeling algorithm with the entire collection of pages.

Table 3 illustrates the relevant topics defacers used from 1998 to 2016. We observed that early defacements (e.g., 1998–2004) were focused on exposing the weak security of the target site with their use of 'security', 'backup', and 'lame.' In 2005, terms such as 'pope,' 'terror,' 'country,' 'marocain,' and 'turk' were the top terms. Although we were unable to form any strong conclusion from the most relevant topics in recent years, it appears that modern defacers are invested in real-world events that are widely covered by the media, and they seem to use defacement to express their viewpoints. Examples: The papal conclave in 2005, the Turkish general election in 2007, the Moroccan general election in 2016, and the coup d'état attempt in Turkey in 2016. In a later section, we will show how our analysis approach allowed us to easily draw parallels from this dataset.

Year	Most Relevant Topics
1998	question, student, **security**, number, place
1999	cowboy, *team*, **security**, think
2000	baby, tabloid, people, provided
2001	lord, prime, provided, saved, better
2002	worry, sind, **lame**, care, **encryption**
2003	**backup**, gift, *team*, came, take
2004	best, *group*, micro, look, total
2005	normal, **pope**, time, familia, contact
2006	**terror**, saved, intruder, energy, user
2007	badger, since, high, **turk**, **turkey**
2008	*crew*, speech, warning, saved, *team*
2009	knowledge, acker, *team*, album, **country**
2010	posted, member, protocol, kernel, security
2011	contact, security, village, holding, highlander
2012	saved, contact, *team*, underground
2013	*team*, forgive, security
2014	eagle, *crew*, electronic
2015	clash, king, **terrorism**, visit, alligator
2016	**marocain**, **turk**, steel, anonymous, *team*

Table 3. Most relevant topics per year

In addition to emphasizing a website's security weakness, another common theme that has been observed across different years was the defacers' *sense of belonging* to a team, as expressed in their use of words like "team," "crew," or "group." We show how DefPloreX-NG enabled us to explore defacers' social structure, which resembles that of gangs characterized by strong and longstanding relationships.

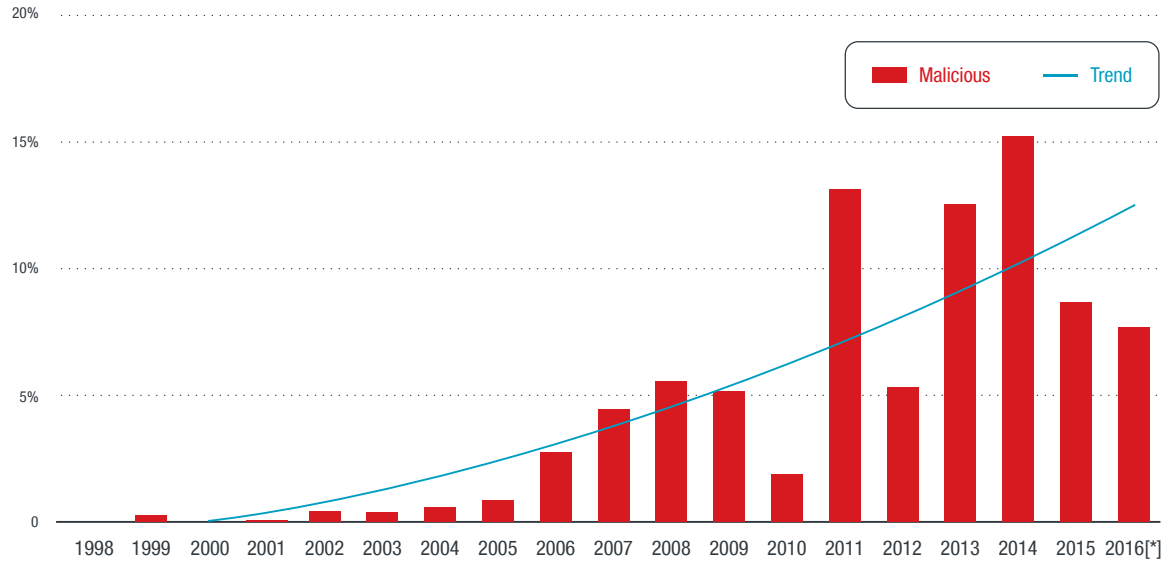
Target Platforms Over the Years

Excluding generic top-level domains (gTLDs) like .com, .org, .net, .edu, .gov, and .mil, the top TLDs are .com.br (4%), .de (3.5%), .co.uk (3.2%), .nl (2.5%), .it (2.3%), and .ru (2.2%). Looking at the breakdown per year (Table 6 in the Appendix), 2002–2009 German websites (.de) were the main targets, while 2010–2016 defacers seem to find most targets in Brazilian (.com.br) and Russian (.ru) sites. Note: Sites can be defaced deliberately (for example, actors can target national or .gov sites for their anti-government sentiments), or they can simply be victimized by chance (for example, unpopular sites with weak security).

In web applications, WordPress, Joomla, and Drupal were frequently targeted by automated exploit scripts whenever a new vulnerability arises. Unsurprisingly, the most targeted platforms are Linux (72 percent), with Windows 2003 (12.3 percent) and 2000 (3 percent) getting second and third places. Although these breakdowns were derived from less trustworthy attribute, these values are still aligned with server-OS usage statistics.[13]

Malicious Content

As shown in Figure 2, there is an increasing trend in the adoption of malicious client-side content (e.g., JavaScript) in deface pages. We observed this trend processing our dataset with Trend Micro™ Site Safety Center,[14] a service that can detect a wide range of web threats, including obfuscated and complex payloads.



Note: The asterisk indicates that in 2016, the data includes only incidents up to September 2016.

Figure 2. Malicious content in deface pages have become prevalent over the years

Given the historical nature of our dataset, and the fact that deface pages are ephemeral, we can only speculate on the reasons behind this trend. The attackers may have used an already-malicious page, or that it might have been targeted with a malicious payload. It's also possible that defacers wanted to monetize their deface page by selling "installation services" to third parties.

Key Observations

Teams and Campaigns

In the psychological analysis conducted by Hyung-jin Woo, Yeora Kim, and Joseph Dominick in 2001, it was posited that defacers cooperate in teams. If driven by strong ideologies, a defacer would not embark in an attack alone but in coordinated fashion with others, the modus operandi resembling that of well-organized cybercriminal gangs.

After manually inspecting thousands of deface pages, we were able to confirm that modern defacers were indeed not simply "script kiddies" but actually tended to be affiliated with teams. In addition, we discovered that they organize their defacements into campaigns, which involved multiple target sites and can be repeated over time. Each group may conduct multiple campaigns, and one campaign may be supported by multiple (partnered) groups.

In our analysis, we found out that the names of the teams as well as their members appeared in the content of deface pages. These observations are taken into account, such that our system can automatically map the relationships between campaigns, teams, and actors.

Deface Templates Used in Campaigns

Because of how teams of defacers operate, notable characteristics can be observed in the resulting deface pages, that is, groups used similar "designs" in their digital vandals. In practice, groups reuse a template that each member can personalize based on the target or other factors. For instance, the two webpages shown in Figure 3 look similar at a glance: Both have a black background and used turquoise text, orange headers, and four embedded videos (one is shown in the screenshots while the other three are at the bottom of the webpage).

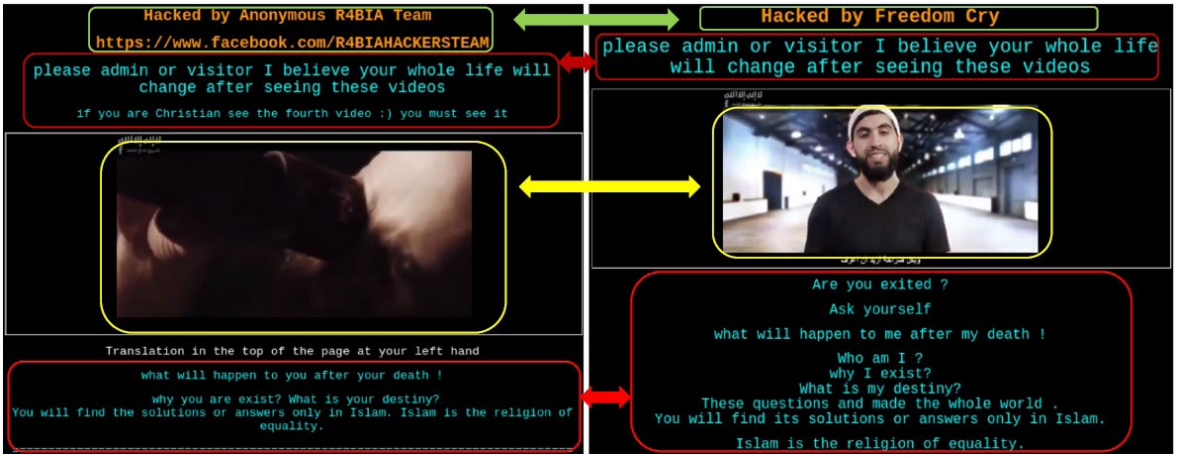


Figure 3. Two deface pages using a similar template
(Anonymous R4BIA Team, left; Freedom Cry, right)

While not always evident at first glance, these two particular deface pages also used the same Western character encoding. Upon manual inspection, it can be concluded that the actor Freedom Cry adopted a template, supposedly created by the Anonymous R4BIA Team, and applied minor personalization.

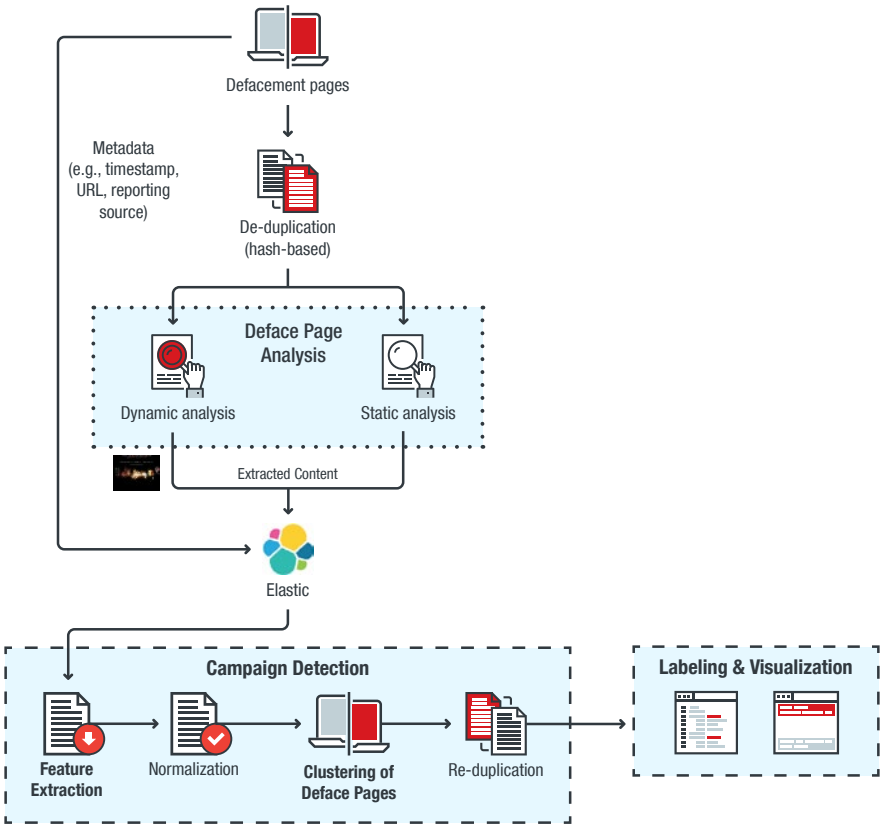
Although the reason and the method behind the reuse of deface page templates are not covered here, we cannot exclude the possibility that novice actors spontaneously copy and reuse templates taken from existing defacements, either to show their willingness to be part of, or even glorify, a defacer team.

Regardless of why, how, and to what extent a page is personalized, our key observation is that deface pages within the same campaign are similar to each other, if not identical. This is a strong attribution indicator, which allows analysts to group deface pages together and understand the relationships between groups and actors. This indicator is the foundation of our approach for automated campaign detection and tracking.

This paper uses the term *campaign template*, or simply template, to indicate the content (for example, bits of text, color scheme, language, character encoding) that is common to most of the pages within a campaign, which in turn can be used to recognize and identify each campaign. However, performing this attribution manually is tedious and extremely time-consuming.

DefPloreX-NG: An Automated Analysis Approach

The previous section dealt with the idea that defacers are organized and act in a way that leaves visible traces of their modus operandi. This section describes how we made use of these traces in order to automatically analyze millions of deface pages to find groups of "similar" pages — that is, pages that belong to the same campaign.



Note: DefPloreX-NG first extracts the raw content from deface pages, both dynamically (in a browser) and statically. Then, we use clustering to detect campaigns as groups of "similar" pages. Lastly, we label each cluster and visualize the campaigns across several dimensions. The core parts highlighted in bold are described in this section while the details are discussed in the next section.

Figure 4. Diagram of our automated analysis approach

Although finding commonalities between webpages is a generic and well-researched problem, deface pages from the same campaign may still differ substantially. Therefore, the concept of "page similarity" is more fragile than in classic information-retrieval settings.

The first phase, called *Deface Page Analysis*, extracts the raw content from deface pages, both dynamically (in a browser) and statically (from the files on a disk). From this content (for example, rendered HTML, images, and other media files), we extract a set of features — the foundation for the remainder of our analysis.

In the *Campaign Detection* phase, we detect campaigns as groups of "similar" deface pages, using an unsupervised machine learning pipeline. More specifically, we use data clustering as the core of our analysis system, where each deface page is an object represented as a tuple of numerical and categorical features. Similar pages will have similar features, and thus will end up being clustered together. Although one might be tempted to rely on a supervised machine learning approach, that is, classification, the lack of ground truth is a showstopper. Our research is rationalized by this lack in any reliable ground truth.

In the last phase, *Labeling & Visualization*, we label each cluster based on its content and visualize the detected campaigns across several dimensions (e.g., time, actors, targets).

In the remainder of this section, we describe the core parts of our approach. The details of the implementation of the approach is discussed in the next section.

Feature Extraction

Engineering the set of features is central to any clustering problem. Based on our experience from manually analyzing thousands of deface pages, we designed the features listed in Table 4 to capture the following aspects: Visual (e.g., color, images, video, audio), structural (e.g., HTML tags), lexicon (e.g., distribution of character classes), social (e.g., use of social network handles), and so on.

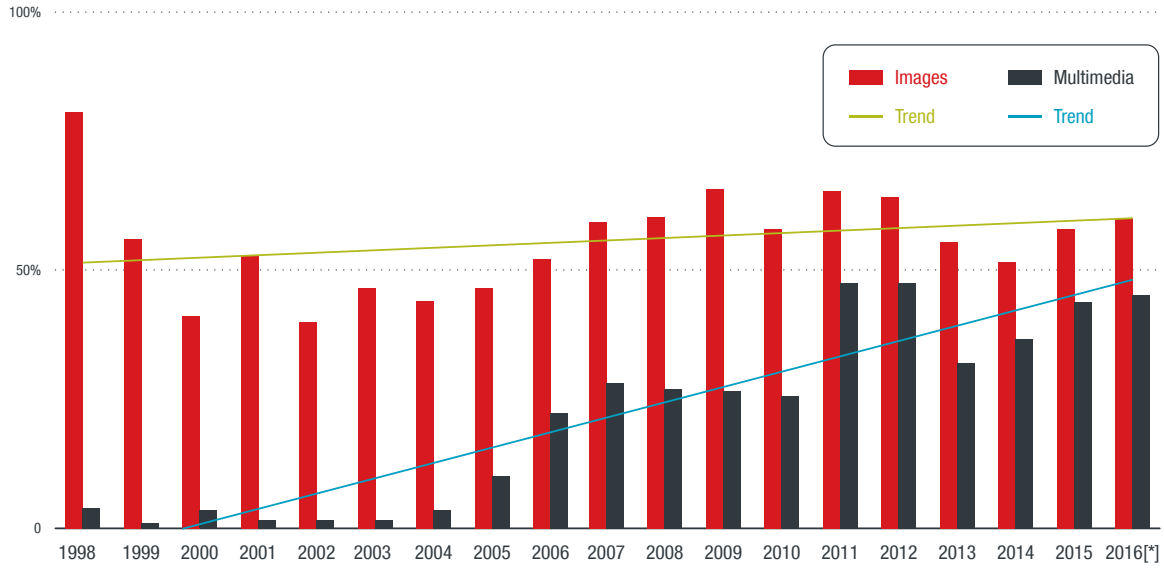
These features are extracted both statically (with pattern-matching on the source files of the deface pages) and dynamically [from the Document Object Model (DOM) obtained through a headless browser].

Although these features are, technically, extracted from attacker-supplied data (i.e., the deface page itself), the fact that in our approach we extract several aspects and that we do not trust the metadata makes our approach more resilient to feature evasion than existing approaches that are purely based on metadata.

Visual Features

As shown in Figure 3, the appearance of a page can immediately characterize a campaign. This has been confirmed by a previous study (see Reference [7]) that showed how a deep learning algorithm finds that small portions of a screenshot are strong features that can automatically tell defaced and clean pages apart.

We extend this concept and include the perceptual hash of the page screenshot, the five most common web-safe colors, and the number of images (that is, <img> tags) found in the page. As shown in Figure 5, web defacers have always used images.



Note: The asterisk indicates that in 2016, the data includes only incidents up to September 2016.

Figure 5. Use of multimedia and image files in deface pages

Furthermore, the data represented in Figure 5 suggests that modern defacers include embedded audio files that play a song whenever the deface page is rendered. Songs are typically related to strong symbols or ideologies which defacers want to promote. These audio files are included via external URLs (for example, pointing to YouTube, SoundCloud, or other streaming services) using JavaScript or the <*embed*> tag.

In our approach, we use two features: The first is numerical and counts the occurrences of "sound URLs"; the second is categorical and captures the type of "sound URLs" such as <*service*>*_srv* (e.g., YouTube, SoundCloud, etc.) and <*ext*>*_file* (e.g., MP3, M4A, WAV, and so on).

Structural Features

The same visual aspect can be obtained through several different combinations of HTML tags. Therefore, we use a set of features that count the occurrences of each tag, focusing on the most relevant tags found in deface pages: style, meta, embed, and object, script, iframe, and a.

Group	Feature Name	Type and Range	Description
Visual	No. of Images	integer [0,∞]	Number of <img> tags
	Perceptual Hash	binary (64 bits)	Calculated on the north-centered 1600x900 screenshot crop
	Average Color	3 floats (RGB)	Average of the five most common colors in the screenshot
	No. of Sound URLs	integer [0,∞]	Number of URLs pointing to sound-hosting services or files
	Type of First Sound URL	categorical	File and service type of the first (usually the only) sound URL
Structural	No. of each <tag>	7 integers [0,∞]	Number of style, embed, script, meta, object, iframe, and a tags
Geographical	Encoding	categorical	Detected text encoding
	Language	categorical	Detected language (for labeling only)
Domains	External Domains	real [0,∞]	Ratio of links pointing to cross-origin domains
	Letters in External Domains	real [0,∞]	Average ASCII letters in the external domains string
Social	No. of Online Handlers	int [0,∞]	Twitter® @handles, #hashtags, email addresses
Title	Letters, Digits, Punctuation, White spaces in Title	4 real [0,∞]	Ratio of the listed character classes in the page title

Table 4. Clustering features that we extract from each deface page

Geographical Features

This group of features captures the ethnicity of the page author. Character encoding is a good indicator of the region (real or mimicked) where the author creates web defacement content. For example, ISO-8859-1 is popular among computer users in Western Europe, ISO-8859-6 in Arabic-speaking regions, and so on. Our approach does not trust the encoding declared with the <*meta*> tag, if any, because some defacers deliberately use an encoding that differs from the declared one, possibly in an attempt to confuse automatic analyzers. Instead, we use the popular chardet library, which is also used by modern web browsers like the Mozilla® Firefox® browser.

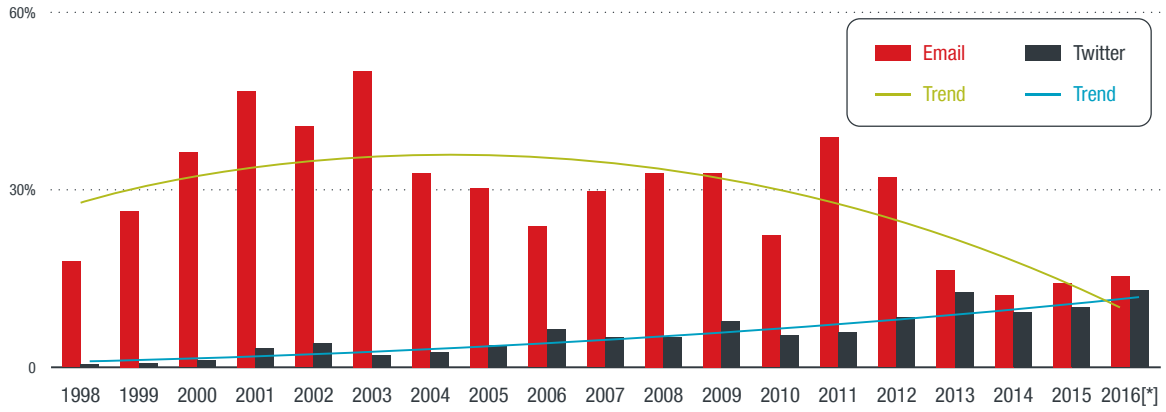
Moreover, the language the text is written in is also detected. Although language detection has become a common task and many reliable tools can perform that task, the co-presence of multiple languages in the same deface page can confuse such tools. Moreover, the typically long list of nicknames creates long sentences of existing and non-existing words in mixed languages, making language detection prone to errors. For this reason, although we have a language-guessing step in our pipeline, we used this feature for descriptive purposes, not for clustering nor to take any decision.

Domain Features

The inclusion of external resources characterizes how a page is built (for example, using libraries or pointing to external URLs versus a self-contained page with embedded resources). This aspect is captured with two features. The first is the ratio of external domains, which is the fraction of URLs included in the page that point to domains that are different from the defaced one. The second feature describes the “syntax” of external domains, that is, the fraction of ASCII letters in each domain name. We keep the average of the latter value for each page.

Social Features

Defacers tend to follow the evolution of internet technologies and adopt mainstream communication and social networking tools, including Internet Relay Chats (IRCs) in earlier years. As shown in Figure 6, the number of Twitter handles and email addresses over time present two opposing trends: The former’s numbers eventually caught up with the latter’s, whose numbers have hugely decreased in later years.



Note: The asterisk indicates that in 2016, the data includes only incidents up to September 2016.

Figure 6. Use of email and Twitter handles in deface pages

Page Title Features

The title is a key element of a webpage, and the same is true for deface pages: The actors use a title that embodies their core message, in a form that catches the attention of search engines and automated scrapers in order to ensure wide visibility. By manually analyzing thousands of deface pages, we noticed that defacers seemed to actually put a reasonable amount of effort in fitting their high-level message into the deface page's title. Sometimes, team member names were also included.

This group of features captures an approximated representation of the lexical aspects of the title, encoded as the ratio of each main character family (ASCII letters, punctuation, white spaces, and digits), and normalized to the title length.

Clustering

Since our dataset contains millions of records, each represented by tens of features, the choice of the clustering algorithm is constrained by the available memory and time. Therefore, any clustering algorithm that needs to materialize the entire distance matrix — or that needs to perform pairwise comparisons across all the combinations of elements — is not a suitable choice. The state of the art for memory efficient, scalable clustering is BIRCH (balanced iterative reducing and clustering using hierarchies).[15]

Instead of calculating and storing the distance between points according to the *entire* feature space, BIRCH keeps three statistics for each cluster: the number of elements in the cluster, the sum, and the square sum. These values are efficient to compute and are sufficient to calculate the distance between two clusters, their centroids, and diameters. Moreover, BIRCH maintains a B+-tree-like structure, which allows a quick searching for the closest cluster for each new data point (that is, deface page feature vector), and updates the tree as new points come in. BIRCH requires one main parameter, the threshold,

which is used to decide whether a new sample should be merged into an existing cluster or if it should start a new one. The branching factor of the tree, which can be tuned, only influences the speed and memory requirements, without drastically affecting the final result.

Optionally, BIRCH clusters can be further post-processed with any other clustering method. However, we manually validated the results with BIRCH alone, and the clusters produced were correct, containing tightly clustered data points, and overall matching the output of the manually clustered data.

While libraries and computation services help streamline machine learning tasks nowadays, we still have to deal with the peculiarities of the application domain.

Categorical Features

Since BIRCH requires all features to be real-valued or, at least, numerical, we reduced the number of categorical features to the bare minimum, with each categorical feature taking a small number of category values. This allows the mapping of each categorical type onto *M*-sized binary features (valued zero or one, accordingly), where *M* is the cardinality of the categorical features. This procedure, known as one-hot encoding, has the downside of increasing the number of features by a factor *M* for each categorical feature.

In Table 4, we one-hot encoded the type of first sound URL (21 categories) and character encoding. The language feature is not used for clustering.

To keep the problem feasible, we reduced the cardinality of the encoding feature to 10 macro-categories. In fact, 40 character encodings exist, which would mean 40 additional binary features. To this end, we grouped the character encodings by region of use, thereby obtaining 10 values (European, Cyrillic, Greek, Turkish, Hebrew, Arabic, Chinese, Thai, Korean, and Japanese) for this feature in no particular order.

Feature Selection and Weighting

Table 4 shows a selection of the features that we originally designed. We eliminated features with near-to-zero variance because they would have no discriminant power in the clustering process. In our research, we eliminated the counts of the <*resource*> and <*link*> HTML tags.

During our validation (detailed in the section on validation), we noticed that the perceptual hash was too discriminant, causing clusters to break too often for minimal visual variations. Indeed, the perceptual hash can be very sensitive to object replacement (for example, the defacer changes the background image, while the campaign is the same). For this reason, we applied feature weighting, assigning 30 percent weight to the perceptual hash feature and 70 percent to the remaining features. We set this percentage weighting empirically (described in the section on validation), starting with 50:50 and gradually shifting the weight towards the other features.

Distance Metric

From the different features, we already have a feature vector with six high-level features (visual, structural, geographical, domains, social, title), comprising 22 real-valued features and 95 binary-valued features, that is, 64 (perceptual hash) + 10 (encoding) + 21 (type of first sound URL). For real-valued features, the Euclidean distance (L2-norm) is the natural choice, whereas binary-valued features are typically compared using the Hamming distance, which is very time efficient. Note: The pairwise Hamming distance between perceptual hashes is a real value in [0,1], where 0 indicates that two images are essentially identical, and 1 indicates that two images are far from being visually similar.

Labeling and Visualization

To provide analysts with an explainable and human-readable view of the clustered deface pages, each cluster is represented as a concise report that includes the time span (oldest and newest deface page), thumbnail of the screenshots grouped by perceptual hash, and, most importantly, a list of patterns that create a meaningful label of that cluster.

To this end, we rely on a set of regular expressions that we built semi-automatically by inspecting thousands of deface pages and clusters. Through these regular expressions, we can reliably extract the name of the deface actor, team, and the set of terms used by the attackers to name their campaign. For example, we built patterns that capture all the variations of sentences such as "*hacked by TEAM / ACTOR NAME*" or "*#CAMPAIGN NAME*," or "*ACTOR NAME defacer*" (including l33t-speech normalization, removal of unnecessary punctuation, and so on). Although far from perfect, extracting these strings semi-automatically from each cluster allows the assignment of a meaningful name to an *otherwise unidentified* set of similar pages, therefore considerably speeding up the manual validation process.

Lastly, each cluster is annotated with the list of categorized targets (e.g., news site, government site), which were obtained from the Trend Micro Site Safety Center.

Armed with these additional attributes, analysts can easily explore, drill in, and pivot the data (demonstrated in the section on validation). For example, we can flexibly group the clusters by labels, period, target (e.g., TLD, category), and so on.

A note on the concept of campaigns: We acknowledge that the concept of a campaign is fuzzy and comes with some limitations. Clusters are meant to find very similar — with respect to the defined features — yet not identical, deface pages. Campaigns are meant as a higher-level grouping of clusters, based on patterns provided by the analyst. In our experience, grouping solely on such patterns does not eliminate the burden of inspecting the large number of similar-yet-not-identical pages. Similarly, clustering alone does not provide semantic labels for the analyst to understand, at a glance, the content of a cluster without inspecting it. For this reason, the definition of campaign encompasses both aspects.

Implementation

In this section, we describe the essential implementation details that are needed to reproduce our data processing pipeline.

Static and Dynamic Page Analysis

The features are extracted both statically (that is, from the page's raw HTML and the page's source file), and dynamically from the DOM that we obtain by rendering the page in a full-fledged headless browser. Technically, the recent Google™ Chrome™ browser "headless browser mode" is the fastest and most robust choice.

For a given feature, when these two static and dynamic values are different, the greatest of the two is kept; we assume that a greater value unveils more content (for example, obfuscated content is revealed during rendering and in the dynamic analysis only).

Despite the fact that defacers are interested in creating functional pages that can be rendered by a browser, the main challenge is that external resources could become unavailable at the time of analysis (for example, HTTP 404 response code). To overcome this limitation, we set a rendering timeout of 10 seconds before storing a partial, but valuable screenshot, and DOM representation.

Deduplication and Reduplication

Often, defacers resort to reusing the same page, resulting to near if not exact duplicate records. We take advantage of this fact to increase the throughput the analysis system is capable of. Indeed, from a clustering viewpoint, two identical pages are treated as one, single page. Therefore, before any expensive computation — from Feature Extraction onward — a hash of the main file is calculated from the deface page (for example, *index.html*), and is used as a deduplication key. After clustering is done on the deduplicated data, the data in each cluster is deduplicated using the same key, thus obtaining the "expanded" clusters with the full set of original records. We opted for the conservative and most precise choice of SHA over SSDEEP.

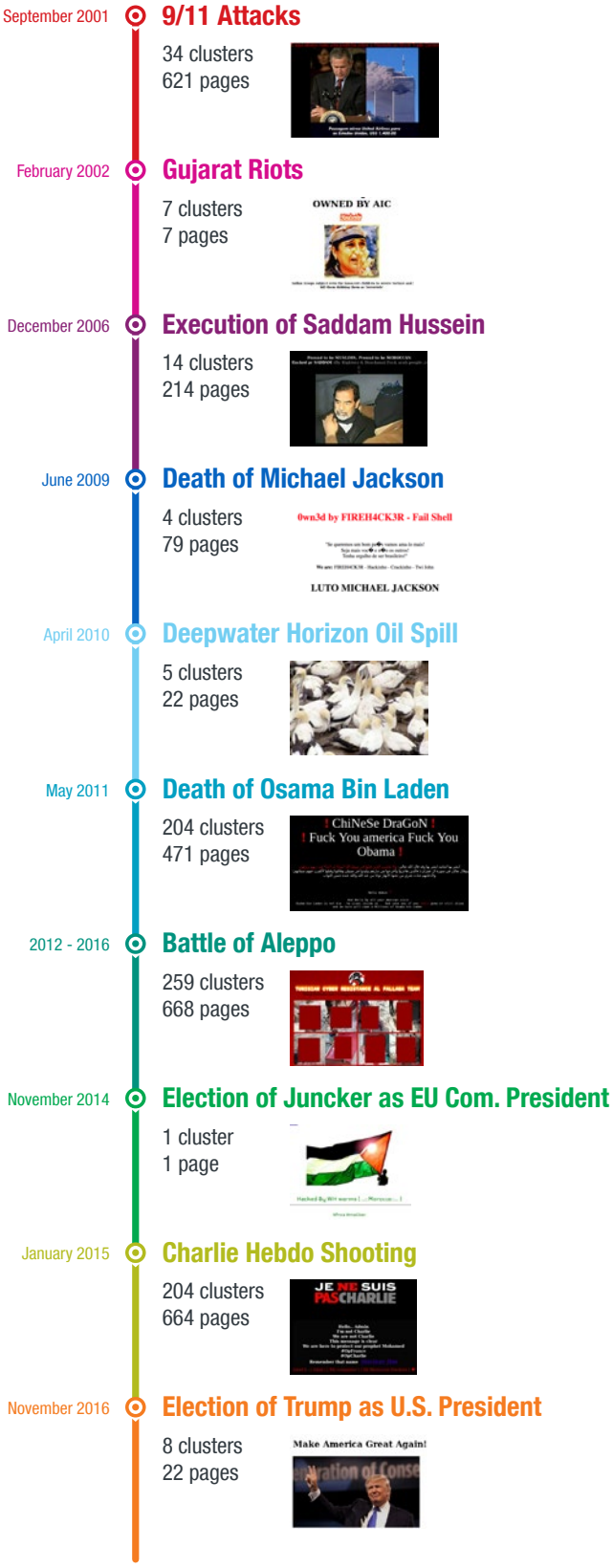


Figure 7. Timeline of major real-world events reflected in web defacement attacks

Normalization

We experimented with a time-dependent data scaling — using yearly min-max bounds — rescaling each feature on a per-year basis, but we obtained very poor results. To understand this outcome, consider Twitter as an example. Before 2006, when Twitter was founded, the value of this feature was essentially zero — apart from defacers who used the *@name* syntax to highlight keywords. Had we used a time-dependent scaling, a value of "one Twitter handle" in 2006 would certainly be the highest value found around that time, whereas the same value in 2016 would be abysmal. As a result, low values would be penalized in recent times, and inflated in the old times, resulting in an overall flattened feature space. In terms of clustering, looking at this sample feature only, a page from 2006 having only one Twitter handle would be clustered together with a page from 2016 having a dozen Twitter handles, because the two values (1 and 12) would have a similar normalized (low) value. After this analysis, we decided not to perform time-dependent scaling and simply resort to dataset-wide normalization using an L2-norm function, which is a widely accepted practice in the machine learning community.

Notes:

- We do not scale binary-valued features, because the Hamming distance does not require any scaling by design.
- The BIRCH clustering algorithm is implemented in Scikit-learn,[16] the de-facto choice for Python-based machine learning. It is well integrated with Pandas and NumPy[17] for data manipulation.

Validation

Validating large-scale clustering results is difficult. Without a ground truth, the only option would be to use *internal* validation metrics, which are computed over the feature space and characterize, for example, how *close* the elements within a cluster are to each other versus how far the elements of two separate clusters are. These metrics do not predicate on the actual quality of a clustering, but simply give an indication on the features' discriminant power. For instance, a perfectly good clustering in practice might have decent internal metrics, and vice versa. Moreover, these metrics do not scale, because they are computed over the pairwise distance.

To overcome these obstacles, we adopted a threefold approach:

- We manually create a small ground truth, comprising 10 defacement campaigns, which we expect our system to cluster accordingly. This allows, first, the calculation of *external* metrics, which indicate how much a clustering "agrees" with the ground truth, and second, to find the best clustering parameters.
- We run our system on a substantial portion (1 percent) of our dataset, and manually inspect the obtained clusters without any pre-built ground truth.
- We run our system on the entire dataset, isolate the largest campaigns that are discovered, project them onto a timeline, and search for confirmatory evidence (for example, online stories) to corroborate such findings.

Initial Validation and Tuning

During our manual analysis of thousands of deface pages, we learned about 10 well-known campaigns, comprising 4,827 pages. We named them Syria.1, Syria.2, Bader Operation, Muslim Liberation Army, End the Occupation, Operation France, Charlie.1, and Charlie.2. .

On this dataset, we ran our system on all the combinations of feature groups (listed in Table 4), with a 0.1 to 0.9 BIRCH threshold. For each iteration, we calculated both internal and external metrics. We used the silhouette score (internal), which is close to 1.0 when clusters are compact and sharply separated clusters. Having a ground truth, we calculated the V-measure, which is the harmonic mean between

homogeneity and completeness. The homogeneity is 1.0 if all the clusters contain only deface pages that are members of a single true campaign, whereas the completeness is 1.0 if all the deface pages of a given true campaign are members of the same cluster. Therefore, a V-measure score equal to 1.0 means a perfect match between ground truth and the obtained clusters.

We obtained the best results with a BIRCH threshold equal to 0.5, resulting in a silhouette score equal to 0.822 and V-measure equal to 0.856 (0.907 homogeneity and 0.810 completeness). Therefore, we set this threshold for the remaining experiments.

We examined what went wrong with few *misclustered* pages, and we learned that those pages did not have an available full screenshot (which affects the visual features). The reason: Some of the external resources were not available anymore. After eliminating such pages from the ground truth and input dataset, we obtained a clustering that matched the ground truth perfectly.

As a final validation, we ran DBSCAN (density-based spatial clustering of applications with noise) on the centroids of the 10 clusters obtained by BIRCH, which were confirmed. Given the time and memory efficiency of BIRCH, we decided to solely use BIRCH.

Large-scale Validation

We validated our system on a larger (but still manually explorable) dataset, which comprised of one month worth of records. We chose a time range, January 2015, where we knew from external sources that there were active campaigns without actually knowing where those campaigns were located in our dataset.

Our system produced 2,722 clusters, of which we inspected 10 percent (totaling 1,702 deface pages). For this, we followed a semi-automated approach with the aid of the *Labeling and Visualization* phase. The screenshots of webpages, keywords, and the *fuzzy* hash (SSDEEP) of the text of webpages helped in speeding up the validation process. However, it shouldn't be a substitute for a complete ground truth.

Through this process, we confirmed the identified campaigns, which included: *op-desaparecidos* (January 10–25) with targets in Mexico and Argentina, *opthailand* (January 5), and *nasaanang* (January 29–31) with targets in the Philippines, including government-related sites.

We did not encounter any spurious cluster (that is, with unrelated defacements). However, we found 27 "split" clusters, despite their content looking very similar. This expected result is due to the conservative approach that favors fine-grained, yet very compact clusters as opposed to spurious ones. Such clusters could be automatically merged during the *Labeling and Visualization* phase.

Real-world Validation

Here, we performed clustering on our whole dataset. Starting with 20 major real-world events, we searched for such evidence in our results. As shown in Figure 7, we found out key matches that confirm the belief that actors use cyberspace as a parallel "territory" for their propaganda.

After the September 11 attacks,[18] pro-Al-Qaeda actors planted supporting webpages celebrating the outcome of the terrorist attacks. Meanwhile, other actors showed opposition to what happened. We observed a similar situation in the 2015 Charlie Hebdo shooting,[19] where both sides of the event gained support from actors.

In February 2002, the riots in Gujarat, India[20] prompted actors to protest against the violence by planting deface pages asking for the end of the riots in compromised Indian sites. We observed a similar reaction during the long and devastating Battle of Aleppo,[21] for which we had to redact part of the screenshot in Figure 7 because of the disturbing images.

As for the events concerning Saddam Hussein's execution[22] in December 2006, and Osama Bin Laden's death[23] in May 2011, several web defacement campaigns showed condemnation of the executors.

Even election events were not spared by defacers. Unidentified actors compromised and defaced the website of the European People's Party (EPP) to protest the election of Jean-Claude Juncker[24] as president of the European Commission in November 2014. In contrast, defacers who supported Donald Trump launched defacement campaigns to celebrate his win in the November 2016 U.S. elections.[25]

Limitations

During our validation process, we discovered that on top of successful cases, there were also clusters with unrelated pages. After manual inspection, we identified two main corner cases that challenged our approach.

One problem pertains to webpages that were defaced with only slight alterations (for example, a short sentence that only says "Admin, you been HACKED! By Attacker"). These yielded feature values with poor discriminant power. Fortunately, such pages seldom form proper campaigns.

The other problem points to defacement that leaves the original, functional web application almost unaltered (for example, webpages that were only planted with traffic-redirection or drive-by code). Such cases do not qualify as defacements. DefPloreX-NG groups deface pages based on the features of the original page, forming clusters containing classes of web applications (e.g., WordPress, Drupal).

Clustering Results

Overall, our clustering took 35 hours of processing time, including 2 hours for *Labeling and Visualization*. In comparison, the fastest alternative for large-scale clustering (that is, DBSCAN) would have required prohibitive and exponentially growing computational resources, as shown in Figure 8. Our attempts to cluster 128,000 deface pages with DBSCAN crashed due to main memory exhaustion of the machine (256GB of RAM). This further confirmed that using BIRCH was the right choice for this type of study.

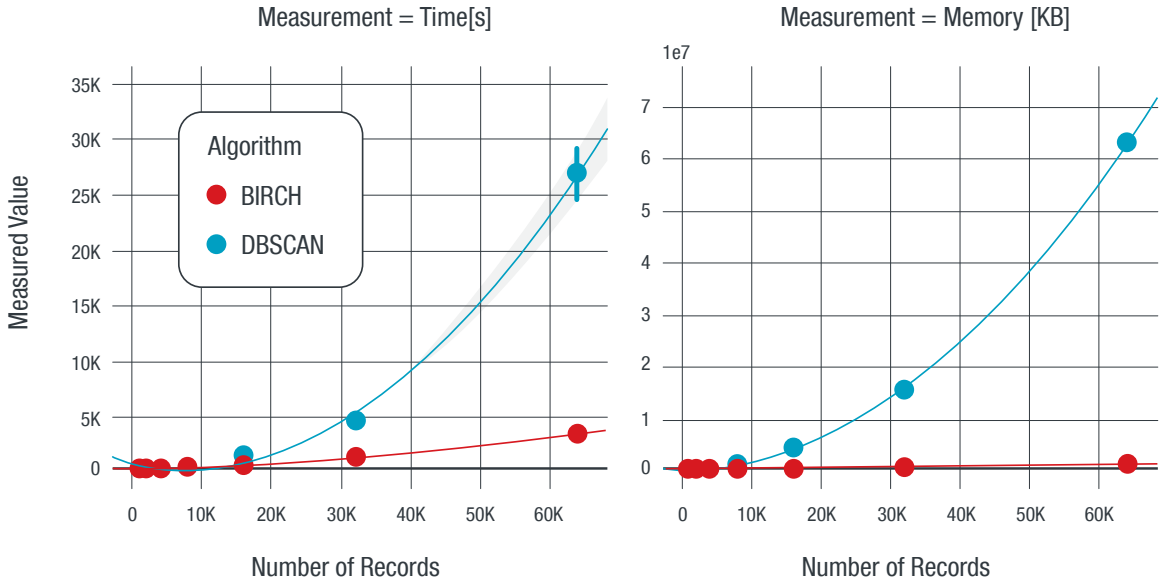


Figure 8. Scalability of BIRCH vs. DBSCAN (10 runs)

Table 5 shows the number of clusters detected by the system over the years, together with the number of teams and attackers. On average, the clusters we obtained included 8–9 records (with a 136–137 standard deviation) and spanned across 9.23 days (with a 77.26 standard deviation).

Year	Actors	Teams	Clusters
1998	50	30	31
1999	410	248	826
2000	820	492	2,385
2001	2,283	1,167	11,726
2002	2,122	1,244	14,684
2003	2,778	2,948	23,183
2004	4,041	4,459	29,722
2005	6,729	5,789	48,043
2006	14,335	9,504	90,632
2007	12,941	7,323	76,000
2008	12,169	6,936	85,085
2009	13,779	8,762	76,567
2010	16,762	8,156	96,599
2011	19,203	8,959	117,396
2012	21,640	10,051	121,243
2013	21,366	10,032	125,195
2014	19,318	8,811	112,760
2015	20,659	12,164	167,031
2016	16,317	10,521	113,085

Note: The 2016 data includes only records up to September of that year.

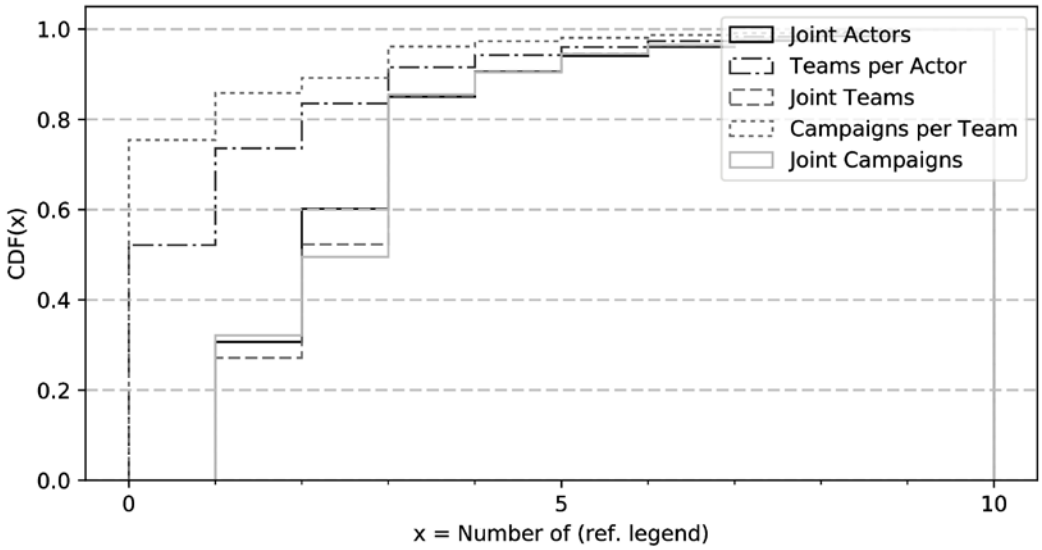
Table 5. Yearly distribution of actors, teams, and clusters as reported by our clustering system (January 1998 to September 2016)

From a practical point of view, each cluster found by our system represents the joint work of multiple actors, working for the same attack, using visually similar deface pages, and, most importantly, sharing the same ideologies. This represents a unique analysis pivot towards a better understanding of how attackers work together, rather than solely relying on anecdotal evidence.

We provide an example of the measurements that analysts can obtain using our system, DefPloreX-NG.

How Attackers Are Organized

First, our system produces insights on how attackers are organized in groups when conducting defacement campaigns. A summary is shown in Figure 9.



Note: In this context, "joint" means "appearing together in the same cluster of deface pages."

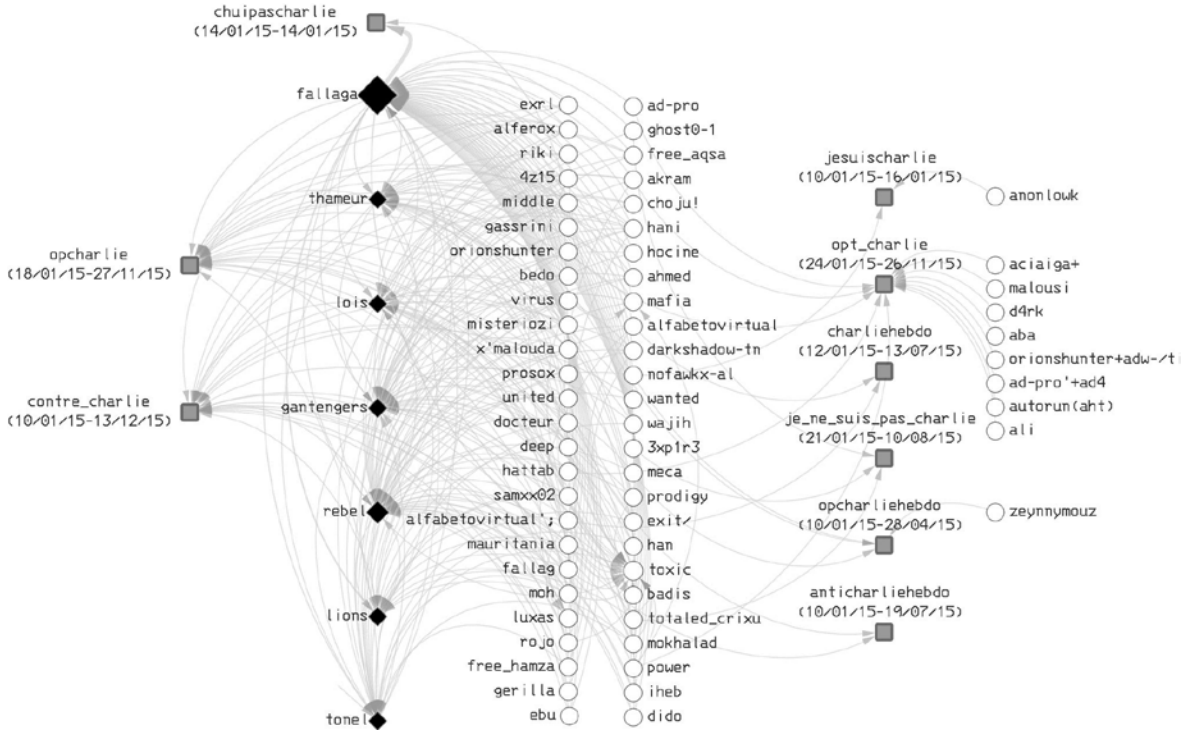
Figure 9. Cumulative distribution of the number of joint actors, affiliations or teams per actor, joint teams, campaigns per team, and joint campaigns

Looking at the whole picture in general terms, about half of the defacers (53 percent) were lone wolves, as they do *not* identify themselves with a team name. The remaining defacers belonged to one or more groups. In addition, the results suggested that most defacers (80 percent) were devoted to the same affiliation/s throughout their career, while only 20 percent migrated from one group to another.

Figure 9 also gives the concept of *joint campaigns* a more elaborate description. Joint campaigns were quite common, with only 30 percent of the campaigns not taking part in the phenomenon. In our analysis, campaigns that shared common motives, objectives, or targets were often driven by similar geopolitical or religious ideologies. Often, they target similar ethnic groups or races. For example, defacement campaigns such as *alepo_se_pierden, savesyria, save_halab, stoptheholocaust, aleppo_is_burning,* and *aleppo_é_in_fiamme*, advocated for the end of the war in Aleppo. These campaigns were operated by hacking groups from countries other than Syria, like Spain, Italy, and others. Actors from anarchist groups such as freedom, OpanArchy, and delirium also operated in joint campaigns to benefit from a wider reach.

Digging deeper into a specific example, Figure 10 provides a view of various campaigns that were pro and against the "Charlie Hebdo" shootings. The Tunisian Fallaga Team[26] participated in the largest of these campaigns ("chuipascharlie" slang for "I am not Charlie"), whereas smaller teams still refer to "fallaga"

in their deface pages. We also noticed a vast majority of actors affiliated with at least one team, and a minority of actors working almost independently.

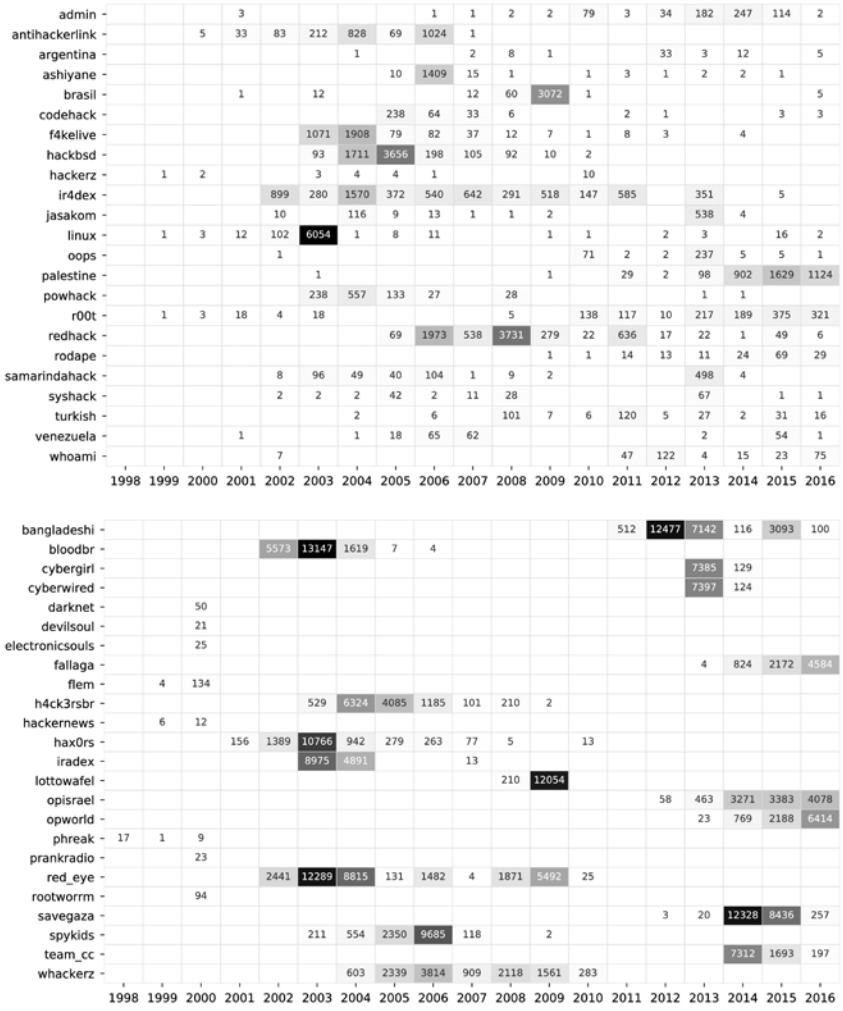


Note: The Fallaga Team drives the largest of these campaigns, while smaller teams (second column of nodes) refer to "fallaga" in their defacements. A small portion of actors (nodes on the right) conduct campaigns almost independently.

Figure 10. Teams (diamond nodes) and actors (circle nodes) cooperate (arrows) in defacement campaigns (square nodes)

Long-term vs. Aggressive Campaigns

Campaigns can differ in terms of durability and intensity. With the following experiment, we captured such differences and visualized them through the heat maps shown in Figure 11.



Note: The heat maps highlight the opposite natures of long-term versus most aggressive campaigns. The cell represents the number of attacks conducted by the campaign per year. Long-term campaigns conduct slower and longer attacks, while aggressive campaigns react to geographical events (such as terrorist attacks) and prefer massive attacks conducted a few days after the event.

Figure 11. Long-term (top) vs. most aggressive (bottom) campaigns

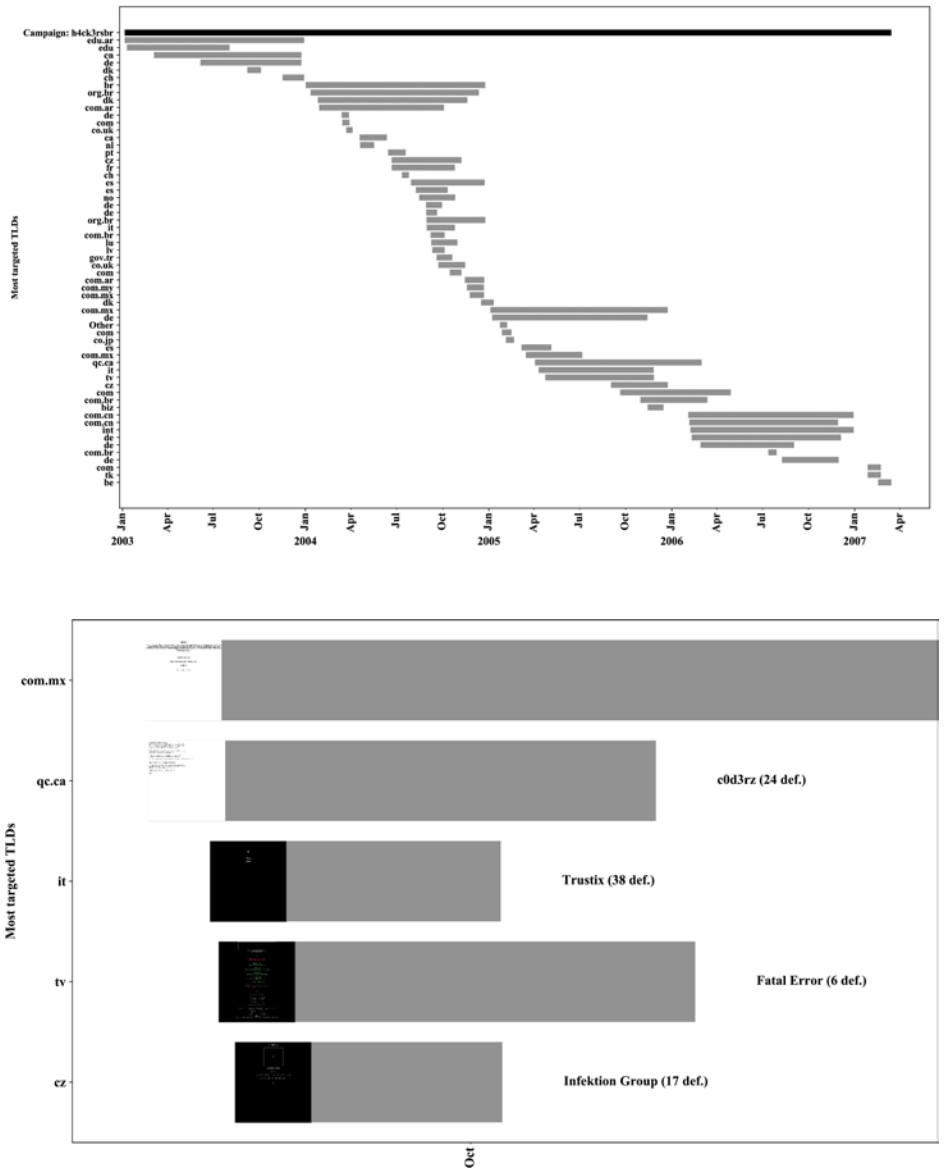
The heat map on the top chart of Figure 11 shows long-term campaigns over our entire dataset. For instance, the *samarindahack* and *syshack* campaigns were present for over 10 years in the underground, with hundreds of attacks distributed over the entire life span of the campaign.

In contrast, the campaigns in the heat map on the bottom chart of Figure 11 lasted shorter, but were more aggressive with respect to the number of attacks they conducted. All of them, including major geopolitical campaigns like *bangladeshi* and *savegaza*, reacted to the conflict in India and Palestine. This implies preference towards visibility over stealth. This aspect is also highlighted in the category of websites that have been targeted, with major news and media portals being defaced for additional visibility.

Case Studies

Throughout this section, we provide examples on how our system can be adopted by analysts to conduct real-world investigations on past and ongoing campaigns.

Timeline Analysis



Note: The campaign's 60 clusters are each represented with a horizontal bar, while highly-targeted TLDs are shown in horizontal axes. A magnified view on the top chart shows five templates (white and black) used by the actors.

Figure 12. The long-running *h4ck3rsbr* campaign

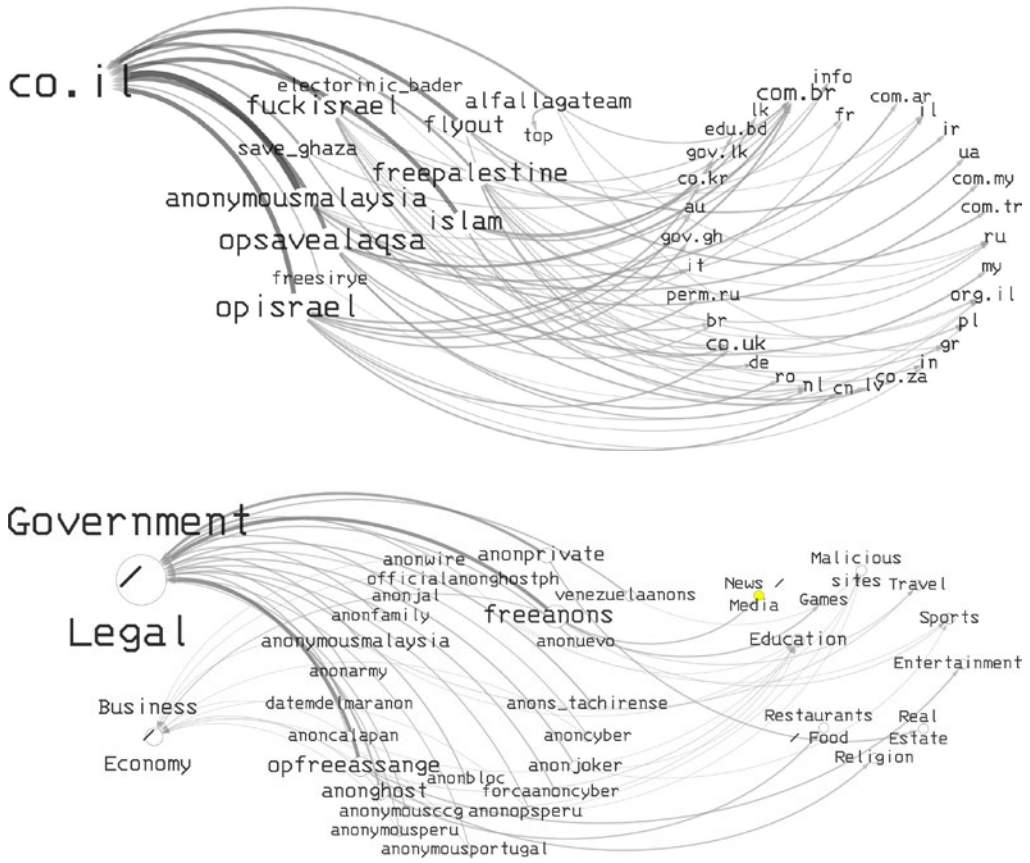
Figure 12 shows a long-running campaign named *h4ck3rsbr*. The 60 clusters are represented as horizontal bars, annotated with the most targeted TLDs (on the vertical axis) and the number of defacements in each cluster (on the magnified detail on the bottom chart).

Although the entire campaign spans over four years, each cluster (that is, horizontal bar) does *not* go over 4 months. However, we were still able to automatically draw a timeline since these clusters share a common label such as the name of the campaign they are affiliated with (that is, h4ck3rsbr).

As opposed to targeted campaigns, h4ck3rsbr is pretty generic, as it targeted websites hosted under various TLDs. Some of the groups that contributed to the campaign were c0d3rz (with a cluster of 24 defacements), Trustix (38 defacements), Fatal Error (six defacements), and Infektion Group (17 defacements). These clusters contain templates with pages having either a white or black background, which were first detected live in October 2006.

Targeted Campaigns: Victims' TLDs and Categories

As previously discussed in the section on how attackers are organized, single campaigns often involve cooperation for the benefit of the entire community. Figure 13 shows that 70 percent of the campaigns cooperate, while nearly half of the joint campaigns are larger than three. As has been said, this is often the case with campaigns sharing common motives and objectives (for example, in support of certain ideologies like religion or politics).



Note: The campaigns cooperating in each group share common motives and objectives: Israeli-Palestinian conflict (top) and Anonymous operations (bottom). The charts show two groups of joint campaigns targeting Israeli websites (top) and government websites (bottom).

Figure 13. Example of cooperation in campaigns

We observed two examples of large-scale joint campaigns in our dataset. Figure 13 visualizes an example of the Israeli-Palestinian conflict (top graph) and Anonymous operations (bottom graph). In each of these graphs, one node is proportionally as big as the number of connecting arcs (that is, defacements). Similarly, one arc is as thick as the number of defacements.

For example, while 12 campaigns were involved in the entire Israeli-Palestine conflict, *opisreal* and *opsavealaqsa* (pertaining to the Al-Aqsa Mosque in Jerusalem) represent the most aggressive and active. With respect to Anonymous operations, these campaigns mainly focused on government websites.

These are only some of the analyses that DefPloreX-NG can provide via automated correlation of the cluster labels.

Conclusion

Attackers compromise and deface websites for various reasons, from promoting their own reputation to, interestingly, promoting a certain ideology or religious or political beliefs. To gain more insight into web defacement, we conducted a large-scale measurement on 13 million records spanning 19 years. Given how the exploration of such a big dataset can take up a large amount of processing resources, not to mention time, we developed an automated analysis approach: DefPloreX-NG. Through this approach, analysts and other security professionals like investigators, penetration testers, and reverse engineers have a convenient and efficient system to help automatically group similar deface pages into clusters, and organize defacement incidents into campaigns. The system uses machine learning algorithms and visualization techniques to turn unstructured data into meaningful high-level descriptions.

For this research, we used the system to understand how defacers collaborate together and how teams are organized to run long-term and aggressive campaigns. We were also able to see how attackers compromise and deface websites for various reasons and observed how campaigns driven by beliefs involve more than one team. The results of our analysis complement if not enrich anecdotal evidence: defacement campaigns that shared common motives, objectives, or targets were often driven by similar geopolitical or religious beliefs and triggered by events relevant to these beliefs.

By demonstrating how an automated analysis system that uses machine learning algorithms can draw trends and correlations from deface pages, we aim to provide cybersecurity researchers and IT teams such as Computer Emergency Readiness Teams (CERTs)/Computer Security Incident Response Teams (CSIRTs) a tool to further study and, in turn, prepare for web defacement attacks and campaigns.

Appendix

Year	Top Five eTLDs	Excluding gTLDs
1998	com co.uk net edu mil	co.uk edu mil it ac.cr
1999	com org net gov se	gov se de edu gov.br
2000	com com.br org net edu	com.br edu gov.br co.uk gov
2001	com com.br net org edu	com.br edu com.tw com.cn de
2002	com net com.br de	com.br de it co.uk
2003	com de net com.br org	de com.br co.uk it
2004	com de net org com.br	de com.br it co.uk nl
2005	com net org de it	de it com.br co.uk ro
2006	com net org de co.uk	de co.uk com.br info nl
2007	com net org de com.br	de com.br info nl co.uk
2008	com net org de com.br	de com.br co.uk nl info
2009	com net org com.br de	com.br de nl co.uk dk
2010	com net org de co.uk	de co.uk nl com.br info
2011	com net org com.br info	com.br info co.uk ru nl
2012	com net com.br org co.uk	com.br co.uk info nl ru
2013	com net org com.br de	com.br de co.uk it ru
2014	com org net ru com.br	ru com.br de co.uk it
2015	com org net com.br co.uk	com.br co.uk ru in it
2016	com net org ru com.br	ru com.br in pl it

Note: The second column is obtained by excluding main generic TLDs (com, org, net)

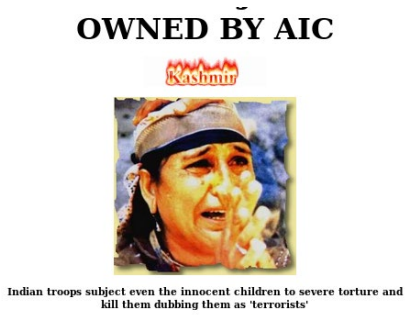
Table 6. Top-level domains targeted each year, based on metadata

Team	Size	Defacements	Nationality
Mafia Hacking Team	47	714,863	Mashhad, Iran
Infektion Group	23	606,309	Brazil
h4x0rteam	31	604,597	Unknown
Hmei7	12	591,388	Indonesia
1923turk	2	547,208	Unknown
red devils crew	15	507,886	Saudi Arabia, China
team falcons hackers	41	491,361	Morocco
nopo team	12	474,379	Iran
ksg-crew	14	406,618	Unknown
anonscorpattackteam	12	356,248	Unknown

Table 7. Top 10 teams overall, according to metadata

Year	Top Five Actors
1998	milw0rm, Team CodeZero, Zyklon, Giftgas, Magica de Bin
1999	AntiChrist, Fuby, PHC, Fl3m, ytcracker
2000	GForce, Prime Suspectz, Hackweiser, pimpshiz, WFD
2001	Silver Lords, BHS, PoizonB0x, Hi-Tech Hate, who
2002	Red Eye, Fatal Error, hax0rs lab, ISOTK, BYS
2003	TechTeam, PsychoPhobia, Red Eye, BloodBR, SHADOW BOYS
2004	iskorpitx, Ir4dex, r00t_System, Infektion Group, int3rc3pt0r
2005	iskorpitx, Infektion Group, ArCaX-ATH, Simiens, SPYKIDS
2006	iskorpitx, Thehacker, crackers_child, CyberLord, SPYKIDS
2007	iskorpitx, 1923Turk, crackers_child, GHoST61, Mafia Hacking Team
2008	iskorpitx, r00t-x, Crackers_Child, GHoST61, Dark_Mare
2009	iskorpitx, NobodyCoder, M0Âµ34d, 1923Turk, Fatal Error
2010	GHoST61, iskorpitx, TheWayEnd, 1923Turk, ByLenis
2011	TiGER-M@TE, 1923Turk, iskorpitx, KriptekS, GHoST61
2012	Hmei7, kinG oF coNTroL, TiGER-M@TE, 1923Turk, T0r3x
2013	Sejeal, Hmei7, misafir, BD GREY HAT HACKERS, SA3D HaCk3D
2014	d3b X, Hmei7, Index Php, Th3Sehzade, 1923Turk
2015	Index Php, w4l3XzY3, Kuroi'SH, d3b X, KingSam
2016	chinafans, GeNErAL, ifactoryx, Freedom Cry, 4Ri3 60ndr0n9

Table 8. Top five actors per year, according to metadata



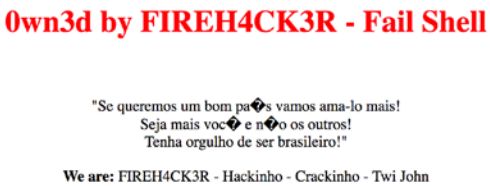


Figure 14. Each screenshot was taken from one of the clusters we identified in our dataset

Related Work

Although website defacement has been a long-standing issue, only limited peer-reviewed works exist. In this section, we divided them roughly between measurements and detection approaches.

Measurements on Web Defacement

H.J. Woo, Y. Kim, and J. Dominick, experts from the community of psychology research, have analyzed the content of 462 deface webpages collected between January and April 2001. They found out that only about 30 percent of defacements had a political motive. Although drawing statistics about the motive of defacements is not part of the goals for this research, we have briefly highlighted in the section *Dataset of Defacement Records* that we have noticed an increased presence of keywords suggesting that defacers are more and more driven by real-world conflicts and ideologies. Interestingly, the same authors also noticed that defacers were not isolated individuals but were part of active and extensive social networks. Unfortunately, 462 hand-picked deface pages were just a drop in the ocean if compared to the data available as of 2009.

The most recent work available, *Hacktivism and Website Defacement: Motivations, Capabilities and Potential Threats*, is very much aligned with our research goal. However, its authors were more interested in figuring out if there is a link (and to what extent) between web defacement and hacktivism. To answer such a question, the authors focused on deface records reported to Zone-H throughout 2016. Interestingly, they also found an increasing number of defacements that are backed by political beliefs and patriotism, especially in the preceding three years. However, the authors relied on metadata (see Table 2), which only shows a portion of the whole picture, rather than on the actual content of the deface page. In addition, it was also very limited in number.

Detection of Web Defacement

Between 2007 and 2008, Giorgio Davanzo, Eric Medvet, and Alberto Bartoli compared the efficacy of seven anomaly detection techniques — versus domain knowledge — at detecting web defacement on a set of 480,000 deface pages obtained from Zone-H. Interestingly, all seven automatic techniques required a feature-space reduction to a dozen features in order to deliver acceptable results. Instead, the use of expert knowledge delivered good results, and sometimes better than the automatic techniques, even with the full array of 1,466 features the authors have selected. This result, extended in their subsequent work in 2011, complements our decision to use a decision-support system as opposed to depending on a pure detection system.

Overall, the problem tackled by Davanzo et al. and Kanti et al is fundamentally different: The authors were looking for features that can recognize defacement in a *monitored* benign page, which is a binary decision. In this research, we were looking for features that can tell the various defacement campaigns apart — a multi-outcome and much more complex decision.

In this direction, Kevin Borgolte, Christopher Kruegel, and Giovanni Vigna used the most advanced technique presented so far. Instead of doing feature engineering and selection, the authors let a deep learning pipeline figure out the best features to recognize deface pages. The input domain to the learning algorithm, which used CNN, was a screenshot of the page and was obtained with a headless browser. As a result, the neural networks automatically give importance to the regions of the page that contain the logos of the hacking groups, or the special “symbols” displayed using uncommon fonts. Indeed, such visual peculiarities can immediately catch the attention of the domain expert, who will find them unusual in a regular benign page. Even though this research went in a different direction, we were inspired by how the authors used the visual appearance of the deface page, thereby leading us to our decision to design some of our features to capture this aspect.

References

1. Kevin Borgolte. *cao.vc*. “A Brief Analysis of the ISIS/ISIL Defacement Campaign.” Last accessed on 5 June 2018 at https://kevin.borgolte.me/notes/team-system-dz-isis-isil-defacement-campaign.

2. John Leyden. (3 November 2014). *The Register*. “Pro-ISIS script kiddies deface West Yorkshire egg-chasers' site.” Last accessed on 5 June 2018 at https://www.theregister.co.uk/2014/11/03/isis_turns_its_hatred_towards_egg_chasers_west_yorkshire.

3. BBC News. (29 January 2014). “Angry Birds website hacked after NSA-GCHQ leaks.“ Last accessed on 5 June 2018 at http://www.bbc.com/news/technology-25949341.

4. Giorgio Davanzo, Eric Medvet, and Alberto Bartoli. A comparative study of anomaly detection techniques in web site defacement detection. In the *International Federation for Information Processing (IFIP) Information Security Conference (ISC)*, pages 711–716. Springer, 2008.

5. Tushar Kanti, Vineet Richariya, and Vivek Richariya. Implementing a web browser with web defacement detection techniques. In *The World of Computer Science and Information Technology Journal (WCSIT), 1 (7):307–310*, 2011.

6. Giorgio Davanzo, Eric Medvet, and Alberto Bartoli. Anomaly detection techniques for a web defacement monitoring service. In *Expert Systems with Applications, 38 (10): 12521–12530*, 2011.

7. Kevin Borgolte, Christopher Kruegel, and Giovanni Vigna. Meerkat: Detecting Website Defacements through Image-based Object Recognition. In *USENIX Security Symposium*, pages 595–610, 2015.

8. UFLDL Tutorial. “Convolutional Neural Network.” Last accessed on June 5, 2018 at http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork.

9. Hyung-jin Woo, Yeora Kim, and Joseph Dominick. Hackers: Militants or Merry Pranksters? A Content Analysis of Defaced Web Pages. In *Media Psychology, 6:1, 63-82*, 2004.

10. Marco Romagna and Niek Jan van den Hout. Hacktivism and Website Defacement: Motivations, Capabilities and Potential Threats. In *Virus Bulletin (VB) Conference*, 2017.

11. *GitHubGist*. “Themes Mass Exploiter Wordpress & Auto Post Zone-h.” Last accessed on 5 June 2018 at https://gist.github.com/dreadpiratesr/798b21f2aa88bc651803.

12. David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (Jan): 993–1022, 2003.

13. W3Techs. “Usage of operating systems for websites.” Last accessed on 5 June 2018 at https://w3techs.com/technologies/overview/operating_system/all.

14. Trend Micro Inc. “Site Safety Center.” Last accessed on 5 June 2018 at https://global.sitesafety.trendmicro.com/.

15. Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: An Efficient Data Clustering Method for very Large Databases. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 103–114, 1996.

16. Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. In *Journal of Machine Learning Research 12: 2825–2830*, November 2011.

17. Wes McKinney. Data Structures for Statistical Computing in Python. S. van der Walt and J. Millman, editors, In *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

18. BBC History. "The 9/11 terrorist attacks." Last accessed on 5 June 2018 at http://www.bbc.co.uk/history/events/the_september_11th_terrorist_attacks.

19. CNN Library. (Updated 25 December 2017). "2015 Charlie Hebdo Attacks Fast Facts." Last accessed on 5 June 2018 at https://edition.cnn.com/2015/01/21/europe/2015-paris-terror-attacks-fast-facts/index.html.

20. The New York Times. (Updated 19 August 2015). "Timeline of the Riots in Modi's Gujarat." Last accessed on 5 June 2018 at https://www.nytimes.com/interactive/2014/04/06/world/asia/modi-gujarat-riots-timeline.html.

21. Reuters. (14 December 2016). "Timeline: The battle for Aleppo." Last accessed on 5 June 2018 at https://www.reuters.com/article/us-mideast-crisis-syria-aleppo-timeline/timeline-the-battle-for-aleppo-idUSKBN1430PJ.

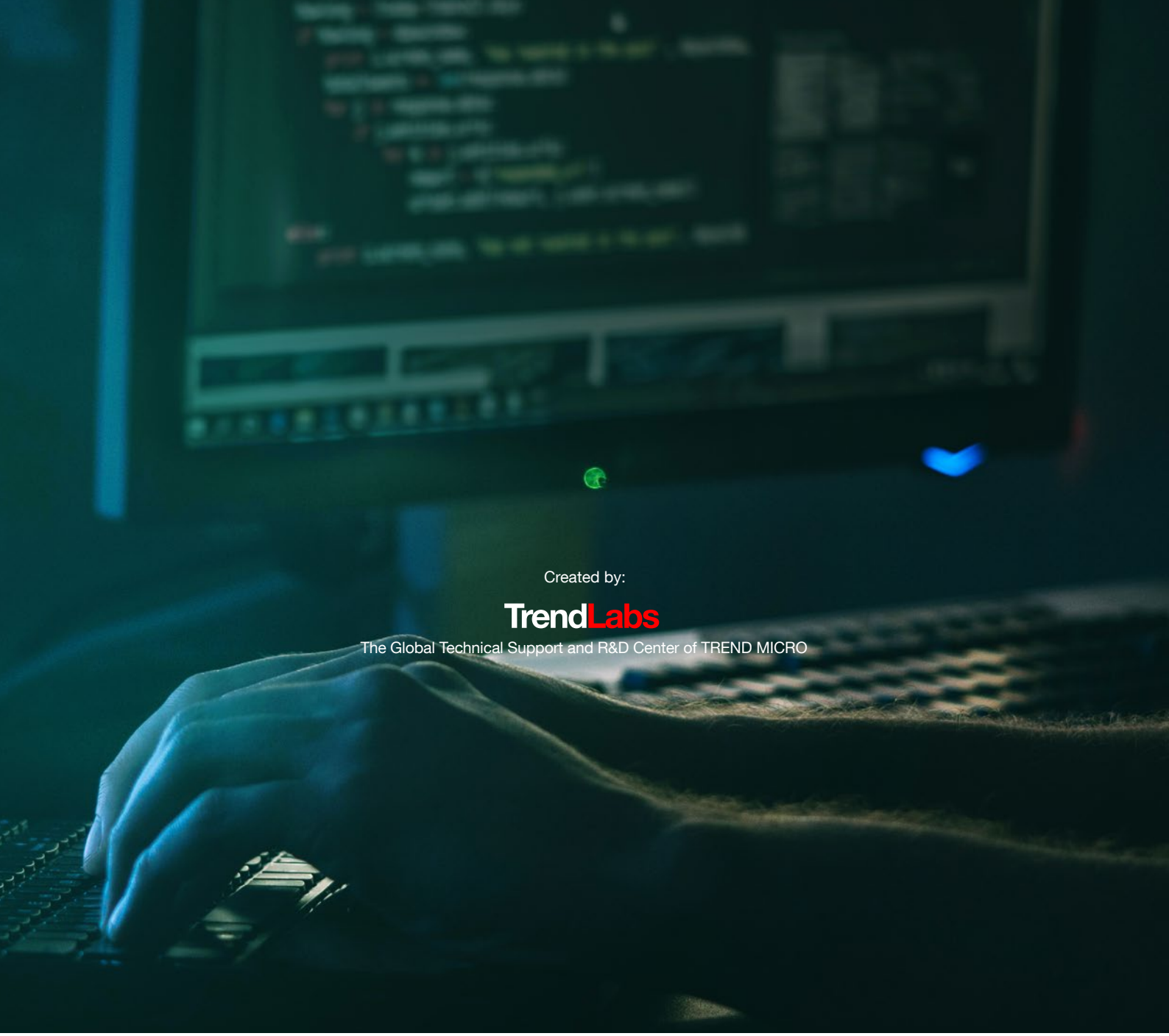
22. CNN World. (30 December 2016). "Hussein executed with 'fear in his face'" Last accessed on 5 June 2018 at http://edition.cnn.com/2006/WORLD/meast/12/29/hussein/.

23. CNN Library. (15 April 2018). "Death of Osama bin Laden Fast Facts." Last accessed on 5 June 2018 at https://edition.cnn.com/2013/09/09/world/death-of-osama-bin-laden-fast-facts/index.html.

24. novinite.com. (15 July 2014). "Jean-Claude Juncker Elected President of EU Commission." Last accessed on 5 June 2018 at https://www.novinite.com/articles/162033/Jean-Claude+Juncker+Elected+President+of+EU+Commission.

25. Matt Flegenheimer and Michael Barbaro. (9 November 2016). *The New York Times*. "Donald Trump Is Elected President in Stunning Repudiation of the Establishment." Last accessed on 5 June 2018 at https://www.nytimes.com/2016/11/09/us/politics/hillary-clinton-donald-trump-president.html.

26. Middle East Media Research Institute (MEMRI) Cyber & Jihad Lab. (5 February 2015.) "Fallaga Team – Tunisian Hacker Group Engages in Jihadi Hacktivism, Active On Twitter, Facebook, YouTube." Last accessed on 5 June 2018 at http://cjlab.memri.org/lab-projects/monitoring-jihadi-and-hacktivist-activity/fallaga-team-tunisian-hacker-group-engages-in-jihadi-hacktivism-active-on-twitter-facebook-youtube.



Created by:

TrendLabs

The Global Technical Support and R&D Center of TREND MICRO

TREND MICRO™

Trend Micro Incorporated, a global leader in cybersecurity solutions, helps to make the world safe for exchanging digital information. Our innovative solutions for consumers, businesses, and governments provide layered security for data centers, cloud environments, networks, and endpoints. All our products work together to seamlessly share threat intelligence and provide a connected threat defense with centralized visibility and control, enabling better, faster protection. With over 6,000 employees in over 50 countries and the world's most advanced global threat intelligence, Trend Micro secures your connected world. For more information, visit www.trendmicro.com.

TREND MICRO™

Securing Your Connected World