Macroscopic view of malware in home networks

Macroscopic View of Malware from Home Networks

Alessandro Finamore*, Sabyasachi Saha†, Gaspar Modelo-Howard†, Sung-Ju Lee†,
Enrico Bocchi*, Luigi Grimaudo*, Marco Mellia*, Elena Baralis*
*Politecnico di Torino †Narus, Inc.
{name.surname}@polito.it {ssaha, gmhoward, sjlee}@narus.com

***Abstract*—Malicious activities on the Web are increasingly threatening users in the Internet. Home networks are one of the prime targets of the attackers to host malwares, commonly exploited as a stepping stone to further launch a variety of attacks. Due to diversification, existing security solutions often fail to detect malicious activities which remain hidden and pose threats to users security and privacy. Characterizing behavioral patterns of known malwares can help to improve the classification accuracy of known threats. More important, since different malwares can share some commonalities, study the behavior of known malwares can enable the detection of previously unknown malicious activities. We pose the research question if it is possible to characterize such behavioral patterns analyzing the traffic from known infected clients. In this paper, we present our quest to discover such characterizations. Results show that commonalities arise but their identification may require some ingenuity. Also, more malicious activities can be found out from this analysis.**

I. Introduction

Malicious activities on the Internet are on the rise. Cyber attackers constantly devise new schemes to evade existing security solutions that typically rely on honeyclients [5], content analysis [4] and blacklisting [8]. As a result, some of these malicious activities remain unnoticed for long period of time, jeopardizing the security and privacy of users behind the compromised end points. Residential networks are even more susceptible to such risks as they are not as protected as enterprise networks. Furthermore, the infected machines can be used by attackers for running malicious campains against other victims.

We aim to characterize the malicious activities in residential networks. Clients in the residential networks are usually equipped with traditional anti-virus tools that are heavily signature or reputation based. They do not have the macro-level view of the traffic and hence could not detect malicious campaign [2]. We have observed that normally some of the malicious attacks are very stealthy and generate very low traffic. But, often infected clients generate traffic that has some commonalities often visible only when looked at the bird's eye view of the traffic in the network. We look into the details of some attacks and try to characterize them. We show that some of the threats have patterns that are easy to detect as they repeat the same traffic over time. But, for others the "pattern" lies in the depth of the traffic.

Existing work characterizes the activities of malware by executing it in an isolated environment and analyzing the traffic it generates. However, sophisticated malware identifies such isolated environment and acts differently [3]. We, on the other hand, characterize the malicious activities from the real traffic. Using network traces collected at a residential network of a large ISP, we explore whether it is possible to identify some meaningful behavioral pattern of suspicious network traffic. We start with the alerts generated by a commercial IDS to detect the malicious events and the corresponding threat label. We then systematically analyze the traffic of clients infected with the same threat looking for common behaviors. If on the one hand the IDS provides hints of where to start the analysis, on the other hand we find ample "hidden" malicious activities that went undetected.

We present our journey to explore the different levels of commonality present in the various threat families. Our contributions are as follow: (i) we present a large scale measurement study of the malware threatening users residential networks, (ii) we identify hidden yet malicious network communications that IDS fails to detect, and (iii) we characterize behavioral patterns of different threat families.

II. Dataset Overview

We consider a vantage point located in a commercial ISP where approximately 20,000 customers are connected. Most of them are residential households, connected via ADSL modems to the monitored point. Each customer's ADSL modem is given a static IP address, which can be used to identify all traffic generated/destined to all terminals used in the household. Since NAT is common, we cannot identify each single device that is connected at each household. In the following, we use the term "user" to generally refer to traffic exchanged by a single household (IP address).

We focus on a trace that we obtained during one day in April 2012. We use a commercial monitoring tool to process the packets in real time and extract a text log file in which each TCP and UDP flow is logged. For each flow, a *record* is stored detailing the classic network tuple (source/destination IP addresses, source destination ports and protocol type), the timestamp of the first packet, the total number of packets and bytes sent and received, and the application protocol used (e.g., HTTP, SMTP, BitTorrent, etc.). When the application protocol is HTTP, the record further reports the server hostname, object path, user-agent, content-type, response status (e.g., `200 OK`), and content-length. In case multiple HTTP transactions are present in the same TCP flow due to HTTP-persistent

TABLE I
DATASET SUMMARY.

Class	All Traffic Users (%)	All Traffic Records (%)	Flagged Traffic Users	Flagged Traffic Records
HTTP	16,217 (*79.1*)	39.7 M (*11.8*)	1,308	42,007
Email	3,640 (*17.7*)	880.7 k (*0.2*)	-	-
Chat	3,045 (*14.8*)	100.8 k (*0.03*)	7	1,467
P2P	3,163 (*15.4*)	17.1 M (*5.05*)	-	-
OthTCP	18,806 (*91.8*)	22.7 M (*6.7*)	24	76
DNS	15,164 (*74.1*)	30.7 M (*9.3*)	-	-
VoIP	8,371 (*40.8*)	80.5 k (*0.02*)	-	-
OthUDP	17,664 (*86.2*)	224.6 M (*66.8*)	-	-
Total	20,486	336.1 M	1,321	43,550

P2P = (eMule, BitTorrent), Email = (SMTP, POP3, IMAP), Chat = (XMPP, YahooMsg, MSN, IRC)

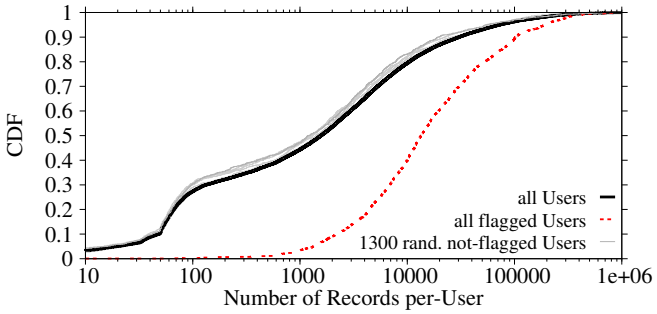

Fig. 1. CDF of the total number of per-user records.

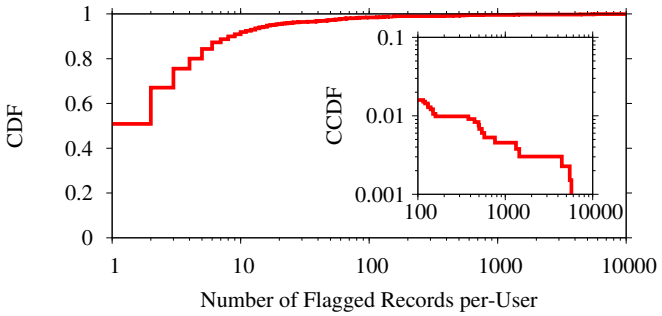

Fig. 2. CDF of the number of flagged records for each user.

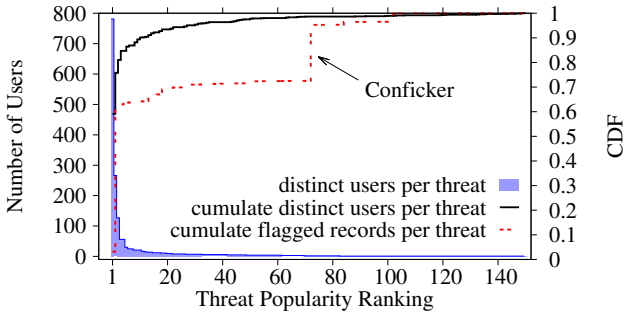

Fig. 3. Overall threats stats.

adoption, multiple records are logged, one for each HTTP transaction. Similarly, for each DNS transaction, the tool logs the requested hostname, the set of IP addresses returned by the resolver, or the response code in case of an error (e.g., not-existent domain). To protect users privacy, sensitive information has been removed and clients IP addresses have been anonymized.

In parallel to the monitoring tool, a commercial IDS processes the packets in real time, producing alerts if a network activity matches any rule that is present in its database. For each alert, the IDS specifies the flow network tuple it relates to and a *threat-ID*, i.e., a numerical code that identifies a particular threat. However, the IDS is very conservative in triggering alerts (possibly) offering a limited view of the network activity related to a malware. Combining the two logs, we obtain the dataset described in Table I. In the following we refer to a *flag* as a log record for which the IDS triggered an alert.

Wide-spreading of Malicious Activities. Overall, 20,486 users generated about 336 million total flows over the whole day. About 20% of those is related to HTTP and DNS records. Surprisingly, a large subset of users (6.4%) exhibit some malicious activity (i.e., at least one record is flagged), with more than 150 different threat-IDs being reported. Yet, only 43,550 flags are raised by the IDS. That translates to a negligible 0.013% of all traffic. Most of these records correspond to HTTP traffic, with the exception of some IRC and RPC flows. This confirms the very stealthy and low rate activity that malware is typically generating. In the following we dig into more details to observe if it is possible to pinpoint differences between infected users (flagged users in the following) and users that are not (not-flagged users).

Traffic Volume of Flagged Users. We first investigate the occurrences at which flagged events happen. The intuition is that the more flagged events occur, the easier should be to spot them in the traffic aggregate. Fig.1 shows the CDF of the amount of total records logged for all user, and for the subset of flagged and not-flagged users (i.e., a random subset of 1,300 users among the not-flagged ones). Results show that the flagged users generate much more traffic than the rest of the population. One would think this is due to the extra traffic generated by the malicious application running at the infected client. However, flagged users present a very small number of flags. This is detailed in Fig. 2 that reports the number of flags per flagged user. We find that 75% (92%) of the users show less than 3 (10) flags in the whole day, and only two users show more than 1,000 flags. As such, while malicious activities can inflate traffic volume, in general the rate of malicious records is very limited. This is a design choice common to many IDS products that prefer to limit the number of reported alarms rather than overloading the analyst with too many events. Unfortunately, this limits also the visibility on the incidents that we observe.

III. MACROSCOPIC VIEW OF THE THREATS

To investigate the threats diversity and the traffic they generate, Fig 3 reports different statistics on the alarms raised by the IDS.

Consider first the shaded histogram. It shows the number of users affected by each threat. Threats are sorted by popularity. Overall, the IDS finds 151 distinct threat activities. Their popularity is highly skewed, with the most popular threat

affecting about 800 (61%) flagged users. However, 129 threats are flagged with less than 10 users each. Despite the limited number of alerts, this highlights a very diversified scenario of malicious activities.

Next, consider the red dashed line of Fig. 3. It reports the CDF of the number of flagged records contributed by each threat. As expected, the majority of these records are related to the most popular threats. However, the distribution has several steps, indicating that some threats are more "chatty" than others and, even when only few users are involved, they still generate many flagged records. For instance, only two users are infected by Conficker [6]; yet they contributes to 23.3% of all flagged records. Note that Conficker is a worm that was first detected in 2008 but is still one of the most popular threats [7].

The solid line in Fig.3 shows the CDF of the number of distinct users involved in each threat. In other words, we progressively add the fraction of new users that were not listed among the flagged-users of previously considered threats. Notice how the distribution presents several "plateaus", indicating that no new flagged users has been added, i.e., all involved users were already accounted by previous threats. We find that 23% of users are flagged with multiple threats. This is due to users being infected by different malwares, and IDS using different threat-IDs to differentiate events for the same class/family/stage of the malware.

Focusing on Popular Threats. We have shown that the number of flagged records per threat varies vastly, due to the number of victims and the frequencies with which the malicious applications generates traffic. We now focus on the subset of threats that exhibit several flagged records, with the goal of characterizing their behavior and finding patterns.

Table II details information on the 15 most popular threats. For some threat-IDs, the IDS provides limited information on the malicious activity and hence we adopt generic names. Notice also how some threats presents a *type*. This corresponds to the ability of the IDS to identify different variant of the network traffic of the same threat. The table reports the number of users affected by each threat-ID, and the associated number of flagged records. Not surprisingly, drive-by downloads and Exploit Kits (EKs) are among the most popular threats. More interesting is the *DynDNS activity* class that corresponds to HTTP requests having hostname registered with DynDNS services that appear to be control messages (e.g., periodic communications to check network connectivity). *Skintrim* and *Tidserv* are two popular trojans that can trigger the download of other malwares through backdoors. *Toolbar activity* threats are related to the *Ask.com* toolbar that are triggered by the download of unwanted advertisement objects or perform iframe injections in the browser.

Table II further details (i) the statistics of users that have more than one flag related to the same threat-ID, and (ii) the average number of flagged records per user. With the exception of DynDNS, it confirms that most of the users exhibit very few events. For instance, the most popular threat is found in 781 distinct users, but 516 of them were flagged only once during the whole day. Moreover, 265 users show more than one flag, but the average is only 3 flags.

TABLE II
MOST POPULAR THREATS.

		Users $\geq$ 1 flag		Users $>$ 1 flag	
#	Name	Users	Flags	Users	AVG flags
1	Drive-by download [type1]	781	1427	265 (33.9%)	3
2	DynDNS activity [type1]	266	26270	181 (68.0%)	144
3	Blackhole EK [type1]	127	158	20 (15.7%)	2
4	Skintrim [type2]	56	301	46 (82.1%)	6
5	Skintrim [type3]	56	301	46 (82.1%)	6
6	Facebook plugin attack	30	31	1 (3.3%)	2
7	Threat-A	25	27	2 (8.0%)	2
8	Blackhole EK [type2]	25	25	- -	-
9	Toolbar activity [type1]	21	105	19 (90.5%)	5
10	Threat-B	21	23	2 (9.5%)	2
11	Threat-C	21	22	1 (4.8)	2
12	Toolbar activity [type2]	17	19	2 (11.8)	2
13	Drive-by download [type2]	15	33	5 (33.3)	4
14	Tidserv	14	228	12 (85.7)	18
15	Threat-D	14	470	7 (50.0)	66

Overall, the frequency and popularity of events is very limited, offering little evidence to investigate some incidents. In the following, we investigate whether we can highlight patterns, at least for those incidents that are popular and repetitive.

IV. SEARCHING FOR PATTERNS IN MALWARES

For threats that have a large number of flagged users and records, we investigate whether there are "patterns", i.e., recurring events that would allow us to identify suspicious activities in new threats. We focus on Threat-D and Tidserv as use-cases. Among all popular threats, these two present the largest number of flags per-user and a manageable popularity (about 10 users each). More importantly, they present different properties that highlight the complexity of the task we try to achieve.

A. Threat-D

Let us consider Threat-D for which we do not have a priori knowledge on the malicious activities it relates to. Fig. 4 shows the whole day traffic evolution over time of the user having the highest number of Threat-D flagged records. Each point corresponds to a different record whose client IP address is the same as the selected user. We focus only on HTTP records. For visualization purpose, we consider an *HTTP event* defined as {server IP address, HTTP object name}. For instance, the records *www.acme.com/main/index.html* and *www2.acme.com/secondary/index.html* that have the same server IP address (e.g., *10.0.0.1*) would be mapped as the same HTTP event "*10.0.0.1/index.html*". Red triangles correspond to the events flagged as "Threat-D"; blue circles are events flagged as threats other that "Threat-D", while gray dots are "not-flagged" events. The result is a compact representation of the traffic evolution over time where points on the same y-values indicate recurrent activity, and vertical shapes indicate different events happening in short amount of time (e.g., the user visiting a new web page and fetching all its objects).

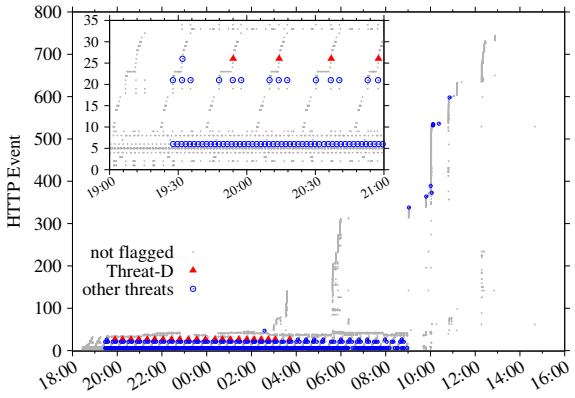

Fig. 4. Evolution of the traffic of the most flagged IP address for Threat-D.

Notice the large periodic activity between 19:30 and 9:00. During this period, the IDS reports several alerts pinpointing specific events as better shown in the figure inset. Table III further details all URIs requested in the first two hours. While the user was (probably) not active (no vertical clusters of point are visible), almost all HTTP transactions relate to the same *instal/file.php* object hosted on different servers (all hostnames map to different IP addresses, thus generating different event). In other words, a sort of "web scan" was happening over night. Further investigating the user-agent string, we correlated this activity to the Ask.com toolbar. We highlight that *instal/file.php* requests represent 78.2% of the overall user's requests. This shows that malicious activity can indeed inflate traffic volume as seen in Fig. 1.

Although the IDS effectively identifies the infected user, only a small subset of suspicious records are flagged. We verified that many suspicious HTTP requests failed (with a response code from the server "404 - Object not found"). Those are not flagged by the IDS, indicating that the IDS rules consider both the client request and the server response.

Digging further among the events that are involved in this incident, we notice a lot of recurrent *www.google.com/webhp* requests. Those are legitimate requests, possibly performed by the malware to run some simple network connectivity check.

Overall, this specific user presents a macroscopic evidence of temporal malicious patterns considering both a strict definition (i.e., specific sequence of events) and a more general one (i.e., repetitive behavior leading to a web scan). IDS provides strong hints of where to start to investigate, but some ingenuity and domain knowledge is needed to further elaborate the analysis.

Unfortunately, we cannot generalize this results. In fact, other 13 users affected by the same Threat-D do not present the same traffic activity. More in detail, only another flagged-user presents one single *instal/file.php* object request. Flagged records instead point to *ads.staticyonkis.com/www/delivery/afr.php* requests happening in short time (up to 12 per second), with no trace of the Ask.com toolbar or connectivity check activities toward Google. In other words, Threat-D exibits to different network behaviors.

TABLE III
7-9 PM REQUESTED URIS FROM THE MOST FLAGGED USER OF THREAT-D.

URI	#
helpindownw.in/instal/file.php	689
notesonacocktailnapkin.com/wp-content/themes/rockstar/instal/file.php	99
www.usedtruckuk.com/wp-content/themes/classic/instal/file.php	93
www.google.com/webhp	73
www.creativelayer.net/modules/mod_wdbanners/instal/file.php	50
advantageclubrockford.com/modules/mod_wdbanners/instal/file.php	50
dreadneck.com/img/instal/file.php	49†
sotobetawi.com/wp-content/uploads/instal/file.php	46
www.veintisietelunas.com/images/stories/instal/file.php	40
www.usmctennis.fr/wp-content/themes/instal/file.php	40
www.radburnd.com/wp-content/themes/desk-mess-mirrored/instal/file.php	40
theuticashale.com/wp-content/themes/spectrum/instal/file.php	40
jiaotong.info/modules/instal/file.php	40
www.jlabmag.com/modules/mod_wdbanners/instal/file.php	25
www.google.com/	13
sas-rep.ru/space	10†
http://gv.t3.idspeed.com/tramp?ref=yonkis0	5
itjustdawnedonme.com/wp-content/uploads/instal/file.php	5
hannaoverstock.com/modules/mod_wdbanners/instal/file.php	5
cuteasabargain.com/wp-content/uploads/instal/file.php	5
bloggasaurus.com/wp-content/instal/file.php	5†
ashishpal.com/wp-content/uploads/instal/file.php	5
www.tpmedia.pl/wp-content/uploads/instal/file.php	4
www.tindellpictures.com/test_blog/wp-content/themes/instal/file.php	4
www.theuticashale.com/wp-content/themes/spectrum/instal/file.php	4
www.symzer.com/hey/wp-content/themes/instal/file.php	4
themarcellusshale.com/wp-content/themes/spectrum/instal/file.php	4
thegenieslamp.net/wp-content/themes/headlines/instal/file.php	4
datemit.com/instal/file.php	4
sasrep.ru/space	3†

† = flagged URIs

To further dissect Threat-D behavior, we collected pcap samples from the vantage point from the client being involved. Those traces reveal the presence of obfuscated javascript code being present in the page. We report an example:

```
/// obfuscated JS code snipped
f=['-32i-32i64i61i-9 ... i10i59i-19i34'][0].split('i');v="e
    "+"va"+"l";}if(v)e=window[v];w=f;s=[];r=String;for
    (;593!=i;i+=1){j=i;s=s+r["f"+"r"+"omC"+"har"+"C"+"ode"
    ](w[j]*1+41);}
[...]

// de-obfuscated JS
document.write("<iframe src='http://gv.t3.idspeed.com/tramp
    ?ref=yonkis0' width='10' height='10' style='visibility:
    hidden;position:absolute;left:0;top:0;'></iframe>")
var f = document.createElement('iframe');f.setAttribute('
    src','http://gv.t3.idspeed.com/tramp?ref=yonkis0');f.
    style.visibility='hidden';f.style.position='absolute';
```

Notice the `f` variable carrying a long vector of chars (not reported entirely for brevity) that corresponds to encoded instructions with respect to an alphabet of symbols contained within the script. Once (manually) decoded, the javascript reveal the `iframe` injection shown in the bottom of the listing. Specifically, those instructions are used to trigger advertisement download (possibly a click fraud activity).

Overall, the lesson learned from analyzing Threat-D is that neither the passive logs nor the IDS alerts were sufficient to identify the real common pattern among the users, i.e., the iframe injection. The IDS groups those users under the same threat and provides the "seed" of when malicious activity happens. Non-negligible ingenuity is required to expand those seeds and spot common patterns.

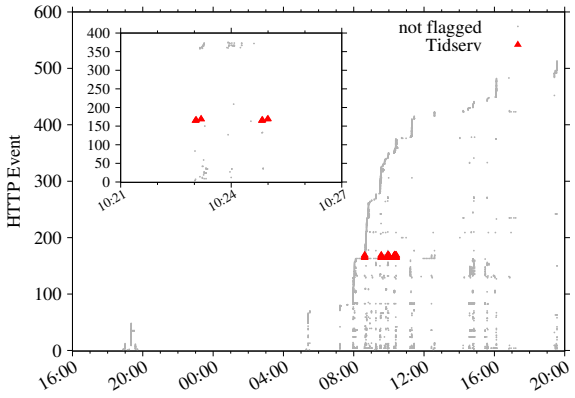

Fig. 5. Evolution of the traffic of the most flagged IP address for Tidserv threat. Patterns are much less evident due to URI and hostname randomization.

B. Tidserv

Consider the Tidserv flagged clients as a second example. Fig. 5 shows the evolution of traffic of the flagged-user presenting the highest number of Tidserv flags. A total of 59 flags are found, mostly happening in short time, as visible by the red triangles. Again, using the sample pcaps and manual inspection, we find this activity enforcing RapidDNS search queries that are related to advertisement download (and possibly click fraud). We report an example:

```
/url = "http://search.rapidns.net/main?InterceptSource=0&
    URI=http%3A%2F%2Fbangl24nj14.com%2F [...]"
if (top.location != location) {
    var w = window, d = document, e = d.documentElement, b
        = d.body,
        x = w.innerWidth || e.clientWidth || b.clientWidth,
        y = w.innerHeight || e.clientHeight || b.
            clientHeight;
    url += "&w=" + x + "&h=" + y;
}
window.location.replace(url)
```

Note the length of the URIs that are longer than 150 characters, presenting a clear randomization and obfuscation component (e.g., *rlyg0-6nbcv.com/Kvb13nWd6P4XrFs3dmVy* [...] *PWZhY2Vib29r27c*). Also in this case, the IDS flags only a subset of the records that exhibit similar patterns, and URIs randomization is successful in hiding malicious events. Note that simply checking URI length or syntactic patterns of the random strings lead to both false negatives and false positives. In general, even when syntactical rules can be spotted, they might not capture the nature of the malicious activity.

V. Discussion & Conclusion

We provided a characterization of malware activities in a real network, starting from a bird's-eye view to discussing technicalities of specific threats. Guided by a commercial IDS, we applied a *bottom-up* approach using the IDS alerts as *seeds* indicating from where to start to look for patterns. By using pcap files to complement the information provided in the dataset, we have finally seen coherent and complete patterns arise. Analyzing such patterns we characterized the behavior of the threats detected by the IDS. Additionally, this identified activities that the IDS failed to detect. However, several issues and challenges remain.

First, using the seeds as a starting point brings the analysis to immediately face the specificities of a threat. This calls for a transparent knowledge of the semantics related to IDS alerts and their role in the overall malicious activities. Commercial IDS solutions might not always provide this information. Open source products such as Snort [1] instead solve the issue by offering public set of rules. However, this comes at the cost of more cumbersome configurations, i.e., without proper knowledge about which rules to select and cathegorize, the system might end up in triggering too false alarms.

Second, despite the strong hints that an IDS provides, those seeds can result in atomic events not easy to correlate with other network events. In other words, by being "too close" to the traffic, we might miss the real general picture related to the network activity of a threat (e.g., rely on obfuscated JS to trigger a web scan or click fraud). Similarly, by trying to extract event sequences or syntactical rules to group HTTP events, we can only end up in too rigid methodologies. Those might capture some regularities but lack the generalization and robustness to malware changes.

Given the complexity reached nowadays by malware, the manual investigation that we performed in the examples does not scale anymore. We strongly believe that the research community has to focus on methodologies to automate the analysis and augment the understanding of a malicious activities found in networks, and especially in home networks where terminals are easily exposed to malware.

How to achieve this is an open research challenge. We believe that a better knowledge of the network dynamics of the threat is needed. However, rather than trying to capture complex threat technicalities within low-level rules, such knowledge should guide the creation of more high-level *malicious symptoms*. Those include atomic events (e.g., executable download) as well as more complex relations among multiple events (e.g., a DNS scan triggering a click fraud). On the other hand, each *symptom* might not be sufficient to identify a malicious activity. It is indeed through the combination of such *symptoms* that we aim to provide a better and more automated methodology to capture malicious patterns.

References

[1] Snort. https://www.snort.org.

[2] L. Invernizzi, S. Miskovic, R. Torres, S. Saha, S.-J. Lee, M. Mellia, C. Kruegel, and G. Vigna. Nazca: Detecting malware distribution in large-scale networks. In *Proc. of IEEE NDSS*, 2014.

[3] A. Kapravelos, M. Cova, C. Kruegel, and G. Vigna. Escape from monkey island: Evading high-interaction honeyclients. In *Proc. of DIMVA*, 2011.

[4] C. Kolbitsch, B. Livshits, B. Zorn, and C. Seifert. Rozzle: De-cloaking internet malware. In *Proc. of IEEE Security and Privacy*, 2012.

[5] J. Nazario. PhoneyC: A virtual client honeypot. In *Proc. of USENIX LEET*, 2009.

[6] P. Porras. Inside risks: Reflections on conficker. *Communications of ACM*, 52(10), Oct. 2009.

[7] L. Seltzer. Conficker: Still spamming after all these years. http://www.zdnet.com/conficker-still-spamming-after-all-these-years-7000031206/, 2014.

[8] J. Zhang, P. Porras, and J. Ullrich. Highly predictive blacklisting. In *Proc. of USENIEX Security*, 2008.