Measuring PUP Prevalence and PUP Distribution through Pay-Per-Install Services

Platon Kotzias, *IMDEA Software Institute and Universidad Politécnica de Madrid;*
Leyla Bilge, *Symantec Research Labs;* Juan Caballero, *IMDEA Software Institute*

https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kotzias

This paper is included in the Proceedings of the
25th USENIX Security Symposium

August 10–12, 2016 • Austin, TX

ISBN 978-1-931971-32-4

Measuring PUP Prevalence and PUP Distribution through Pay-Per-Install Services

Platon Kotzias
IMDEA Software Institute &
Universidad Politécnica de Madrid, Spain
platon.kotzias@imdea.org

Leyla Bilge
Symantec Research Labs
Sofia Antipolis, France
leyla_bilge@symantec.com

Juan Caballero
IMDEA Software Institute
Madrid, Spain
juan.caballero@imdea.org

Abstract

Potentially unwanted programs (PUP) such as adware and rogueware, while not outright malicious, exhibit intrusive behavior that generates user complaints and makes security vendors flag them as undesirable. PUP has been little studied in the research literature despite recent indications that its prevalence may have surpassed that of malware.

In this work we perform the first systematic study of PUP prevalence and its distribution through pay-per-install (PPI) services, which link advertisers that want to promote their programs with affiliate publishers willing to bundle their programs with offers for other software. Using AV telemetry information comprising of 8 billion events on 3.9 million real hosts during a 19 month period, we discover that over half (54%) of the examined hosts have PUP installed. PUP publishers are highly popular, e.g., the top two PUP publishers rank 15 and 24 amongst all software publishers (benign and PUP). Furthermore, we analyze the who-installs-who relationships, finding that 65% of PUP downloads are performed by other PUP and that 24 PPI services distribute over a quarter of all PUP. We also examine the top advertiser programs distributed by the PPI services, observing that they are dominated by adware running in the browser (e.g., toolbars, extensions) and rogueware. Finally, we investigate the PUP-malware relationships in the form of malware installations by PUP and PUP installations by malware. We conclude that while such events exist, PUP distribution is largely disjoint from malware distribution.

1 Introduction

Potentially unwanted programs (PUP) are a category of undesirable software that includes adware and rogue software (i.e., rogueware). While not outright malicious (i.e., malware), PUP behaviors include intrusive advertising such as ad-injection, ad-replacement, pop-ups, and pop-unders; bundling programs users want with undesirable programs; tracking users' Internet surfing; and pushing the user to buy licenses for rogueware of dubious value, e.g., registry optimizers. Such undesirable behaviors prompt user complaints and have led security vendors to flag PUP in ways similar to malware.

There exist indications that PUP prominence has quickly increased over the last years. Already in Q2 2014, AV vendors started alerting of a substantial increase in collected PUP samples [59]. Recently, Thomas et al. [64] showed that ad-injectors, a popular type of PUP that injects advertisements into user's Web surfing, affects 5% of unique daily IP addresses accessing Google [64]. And, Kotzias et al. [35] measured PUP steadily increasing since 2010 in (so-called) malware feeds, to the point where nowadays PUP samples outnumber malware samples in those feeds. Still, the prevalence of PUP remains unknown.

A fundamental difference between malware and PUP is distribution. Malware distribution is dominated by *silent* installation vectors such as drive-by downloads [22, 53], where malware is dropped through vulnerability exploitation. Thus, the owner of the compromised host is unaware a malware installation happened. In contrast, PUP does not get installed silently because that would make it malware for most AV vendors. A property of PUP is that it is installed with the consent of the user, who (consciously or not) approves the PUP installation on its host.

In this work, we perform the first systematic study of PUP prevalence and its distribution through pay-per-install (PPI) services. PPI services (also called PPI networks) connect advertisers willing to buy installs of their programs with affiliate publishers selling installs. The PPI services used for distributing PUP are disjoint from silent PPI services studied by prior work [7]. Silent PPI services are exclusively used for malware distribution, while the PPI services we study are majoritarily used for distributing PUP and benign software. In the analyzed PPI services, an affiliate publisher owns an original program (typically freeware) that users want to install. To monetize installations of its free program, the affiliate publisher bundles (or replaces) it with an installer from a PPI service, which it distributes to users looking for the original program. During the installation process of the original program, users are prompted with offers to also install other software, belonging to advertisers that pay the PPI service for successful installs of their advertised programs.

To measure PUP prevalence and its distribution through PPI services we use AV telemetry information comprising 8 billion events on 3.9 million hosts during a 19 month time period. This telemetry contains events where parent programs installed child programs and we focus on events where the publishers of either parent or child programs are PUP publishers. This data enables us to measure the prevalence of PUP on real hosts and to map the who-installs-who relationships between PUP publishers, providing us with a broad view of the PUP ecosystem.

We first measure PUP prevalence by measuring the installation base of PUP publishers. We find that programs from PUP publishers are installed in 54% of the 3.9M hosts examined. That is, more than half the examined hosts have PUP. We rank the top PUP publishers by installation base and compare them with benign publishers. The top two PUP publishers, both of them PPI services, are ranked 15 and 24 amongst all software publishers (benign or not). The top PUP publisher is more popular than NVIDIA, a leading graphics hardware manufacturer. The programs of those two top PUP publishers are installed in 1M and 533K hosts in our AV telemetry dataset, which we estimate to be two orders of magnitude higher when considering all Internet-connected hosts. We estimate that each top 20 PUP publisher is installed on 10M–100M hosts.

We analyze the who-installs-who relationships in the publisher graph to identify and rank top publishers playing specific roles in the ecosystem. This enables us to identify 24 PPI services distributing PUP in our analyzed time period. We also observe that the top PUP advertisers predominantly distribute browser add-ons involved in different types of advertising and by selling software licenses for rogueware. We measure PUP distribution finding that 65% of PUP downloads are performed by other PUP, that the 24 identified PPI services are responsible for over 25% of all PUP downloads, and that advertiser affiliate programs are responsible for an additional 19% PUP downloads.

We also examine the malware-PUP relationships, in particular how often malware downloads PUP and PUP downloads malware. We find 11K events (0.03%) where popular malware families install PUP for monetization and 5,586 events where PUP distributes malware. While there exist cases of PUP publishers installing malware, PUP–malware interactions are not prevalent. Overall, it seems that PUP distribution is largely disjoint from malware distribution. Finally, we analyze the top domains distributing PUP, finding that domains from PPI services dominate by number of downloads.

Contributions:

- We perform the first systematic study of PUP prevalence and its distribution through PPI services using AV telemetry comprising 8B events on 3.9M hosts over a 19-month period.
- We measure PUP prevalence on real hosts finding that 54% have PUP installed. We rank the top PUP publishers by installation base, finding that the top two PUP publishers rank 15 and 24 amongst all (benign and PUP) software publishers. We estimate that the top 20 PUP publishers are each installed on 10M-100M hosts.
- We build a publisher graph that captures the who-installs-who relationships between PUP publishers. Using the graph we identify 24 PPI services and measure that they distribute over 25% of the PUP.
- We examine other aspects of PUP distribution including downloads by advertiser affiliate programs, downloads of malware by PUP, downloads of PUP by malware, and the domains from where PUP is downloaded. We conclude that PUP distribution is largely disjoint from malware distribution.

2 Overview and Problem Statement

This section first introduces the PPI ecosystem (Section 2.1), then details the datasets used (Section 2.2), and finally describes our problem and approach (Section 2.3).

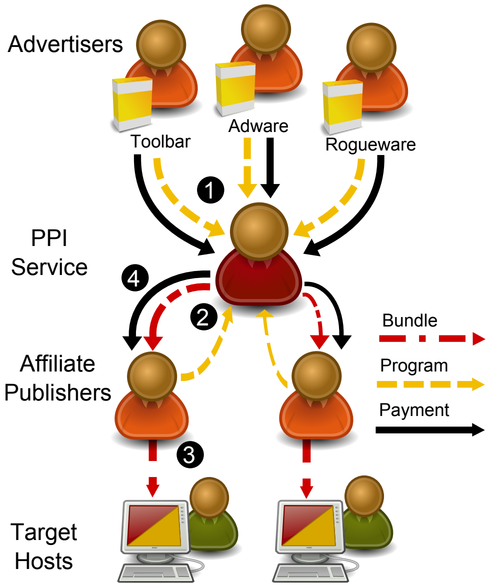

Figure 1: Typical transactions in the PPI market. (❶) Advertisers provide software they want to have installed, and pay a PPI service to distribute it. (❷) Affiliate publishers register with the PPI service, provide their program, and receive a bundle of their program with the PPI installer. (❸) Affiliate publishers distribute their bundle to target users. (❹) The PPI service pays affiliate publishers a bounty for any successful installations they facilitated.

2.1 Pay-Per-Install Overview

The PPI market, as depicted in Figure 1, consists of three main actors: *advertisers*, *PPI services/networks*, and *affiliate publishers*. *Advertisers* are entities that want to install their programs onto a number of target hosts. They wish to *buy installs* of their programs. The PPI service receives money from advertisers for the service of installing their programs onto the target hosts. They are called advertisers because they are willing to pay to promote their programs, which are offered to a large number of users by the PPI service. Advertiser programs can be benign, potentially unwanted (PUP), and occasionally malware.

Affiliate publishers are entities that *sell installs* to PPI services. They are often software publishers that own programs (e.g., freeware) that users may want to install, and who offer the advertiser programs to those users installing their programs. This enables affiliate publishers to monetize their freeware, or to generate additional income on top of the normal business model of their programs. They can also be website owners that offer visitors to download an installer from the PPI service, thus selling installs on the visitor's machines. Affiliate publishers are often referred simply as publishers, but we use publishers to refer to software owners, and affiliate publishers for those signing up to PPI services.

Country	Avg	Range
United States	$1.30	$0.70-$2.00
United Kingdom	$0.80	$0.40-$1.50
Australia	$0.40	$0.30-$0.50
Canada	$0.40	$0.30-$0.50
France	$0.28	$0.15-$0.50
Germany	$0.25	$0.10-$0.40
New Zealand	$0.23	$0.15-$0.35
Ireland	$0.19	$0.15-$0.25
Denmark	$0.18	$0.15-$0.20
Austria	$0.16	$0.15-$0.20
Netherlands	$0.16	$0.10-$0.20
Finland	$0.15	$0.10-$0.20
Norway	$0.15	$0.05-$0.20
Switzerland	$0.12	$0.03-$0.20
Spain	$0.11	$0.03-$0.20

Table 1: Top 15 countries with the highest average price per install collected from 3 PPI services [8,9,50] on June 2016.

The PPI service acts as a middle man that buys installs from affiliate publishers and sells installs to advertisers. The PPI service credits the affiliate publisher a *bounty* for each confirmed installation, i.e., affiliate displays an offer for an advertised program *and* the user approves and successfully installs the advertised program.

Affiliate publishers are paid between $2.00 and $0.01 per install depending on the geographic location. Prices vary over time based on offer and demand and the current price is typically only available to registered affiliate publishers. Table 1 shows the prices paid to affiliate publishers for the most demanded countries on June 25th, 2016 by 3 PPI services that publicly list their prices to attract affiliate publishers. The highest paid country is the United States with an average install price of $1.30, followed by the United Kingdom ($0.80), Australia and Canada ($0.40), and European countries starting at $0.30 for France. The cheapest installs are $0.03–$0.01 for Asian and African countries (typically part of a "Rest of the World" region). In comparison, prices paid to affiliate publishers by silent PPI services that distribute malware range $0.18-–$0.01 per install [7]. This shows that malware distribution can be an order of magnitude cheaper for the most demanded countries.

A common PPI model (depicted in Figure 1) is that the affiliate publisher provides the PPI service with its program executable and the PPI service *wraps* (i.e., bundles) the affiliate's program with some *PPI installer* software, and returns the bundle/wrapper to the affiliate publisher. The affiliate publisher is then in charge of distributing the bundle to users interested in the affiliate's program. The distribution can happen through different vectors such as websites that belong to the affiliate publisher or uploading the bundle to *download portals* such as Download.com [15] or Softonic [61]. When a user executes the wrapper, the wrapper installs the affiliate's program and during this installation it offers the user to install other advertised programs. If the user downloads and installs one of the offers, the PPI service pays a bounty to the affiliate's account.

An affiliate publisher can register with a PPI service even it if it does not own programs that users want to install. Some PPI services look for affiliate website owners whose goal is to convince visitors of their websites to download and run an installer from the PPI service. Furthermore, some PPI services offer a *pre-wrapped software* model where the PPI service wraps its own software titles with the advertiser offers, and provides the bundle to the affiliate publishers [29]. Some PPI services even allow affiliate publishers to monetize on third-party free programs (e.g., GNU).

Some download portals such as Download.com run their own PPI service. When publishers upload their programs to the portal (e.g., through Upload.com) they are offered if they want to monetize their programs. If so, the download portal wraps the original program and makes the bundle available for download. In this model the download portal is in charge of distribution.

Another distribution model are *affiliate programs* where an advertiser uses affiliate publishers to distribute its software directly, without a PPI service. This is a one-to-many distribution model, in contrast with the many-to-many distribution model of PPI services.

2.2 Datasets

Our paper leverages several datasets to conduct a systematic investigation about PUP prevalence and distribution. We analyze WINE's binary downloads dataset [16] to trace PUP installations by real users and their parent/child (downloader/downloadee) relationships, the list of signed malicious executables from the Malsign project [35] to cluster together executables signed by different signers that belong to the same publisher, and VirusTotal [67] reports for enriching the previous datasets with additional file meta-data (e.g., AV detections, file certificates for samples not in Malsign). Table 2 summarizes these datasets.

Dataset	Data	Count
WINE	Events Analyzed	8 B
01/2013 – 07/2014	Events with Parent	90 M
	Total number of Machines	3.9 M
	All Files	2.6 M
	Parent Files	657 K
	Child Files	2 M
	Signed Files	982 K
	Publishers	6 K
	Parent Publishers	1.4 K
	Child Publishers	6 K
	Events with URL	1.1 M
	URLs	290 K
	FQDNs	13.4 K
	ESLDs	7.5 K
Malsign	Signed executables	142 K
VirusTotal	Reports	12 M
	Feed Reports	11 M
	WINE Reports	1.1 M
	Malsign Reports	142 K

Table 2: Summary of datasets used.

WINE. The Worldwide Intelligence Network Environment (WINE) [17] provides researchers a platform to analyze data collected from Symantec customers that opt-in to the collection. This data consists of anonymous telemetry reports about security events (e.g., AV detections, file downloads) on millions of real computers in active use around the world.

In this work, we focus on the *binary downloads* dataset in WINE, which records meta-data about all executable files (e.g., EXE, DLL) and compressed archives (e.g., ZIP, RAR, CAB) downloaded by Windows hosts regardless if they are malicious or benign. Each *event* in the dataset can correspond to (1) a download of an executable file or compressed archive over the network, or (2) the extraction of a file from a compressed archive. For our work, we analyze the following fields: the server-side timestamp of the event, the unique identifier for the machine where the event happens, the SHA256 hash of the child file (i.e., downloaded or extracted), the SHA256 hash of the parent process (i.e., downloader program or decompressing tool), the certificate subject for the parent and child files if they are signed, and, when available, the URL from where the child file was downloaded. The files themselves are not included in the dataset.

We focus our analysis on the 19 months between January 1st 2013 and July 23rd 2014. As our goal is to analyze PUP (i.e., executables from PUP publishers), we only monitor the downloads of PUP and the files that are downloaded by PUP, i.e., events where either the child or the parent is PUP. This data corresponds to 8 billion events. The details of the data selection methodology are explained in Section 3. Out of 8 B events, 90 M events have information about the parent file that installed the PUP. Those events comprise 2.6 M distinct executables out of which 982 K (38%) are signed by 6 K publishers.

A subset of 1.1 M events provide information about the URL the child executable was downloaded from. These events contain 290 K unique URLs from 13.4 K fully qualified domain names (FQDNs). To aggregate the downloads initiated from the same domain owner, we extract the effective second-level domain (ESLD) from the FQDN. For example, the ESLD of `www.google.com` is the 2LD `google.com`, however, the ESLD of `www.amazon.co.uk` is the 3LD `amazon.co.uk` since different entities can request `co.uk` subdomains. We extract the ESLDs of the domains by consulting Mozilla's public suffix list [54].

Malsign. To cluster executables in the WINE binary downloads dataset signed by different entities that belong to the same publisher, we leverage a dataset of 142 K signed malware and PUP from the Malsign project [35]. This dataset includes the samples and their clustering into families. The clustering results are based on statically extracted features from the samples with a focus on features from the Windows Authenticode signature [39]. These features include: the leaf certificate hash, leaf certificate fields (i.e., public key, subject common name and location), the executable's hash in the signature (i.e., Authentihash), file metadata (i.e., publisher, description, internal name, original name, product name, copyright, and trademarks), and the PEhash [68]. From the clustering results we extract the list of publisher names (subject common name in the certificates) in the same cluster, which should belong to the same publisher.

VirusTotal. VirusTotal [67] is an online service that analyzes files and URLs submitted by users. One of its services is to scan the submitted binaries with anti-virus products. VirusTotal also offers a web API to query meta-data on the collected files including the AV detection rate and information extracted statically from the files. We use VirusTotal to obtain additional meta-data about the WINE files, as well as from 11 M malicious/undesirable executables from a feed. In particular, we obtain: AV detection labels for the sample, first seen timestamp, detailed certificate information, and values of fields in the PE header. This information is not available otherwise as we do not have access to the WINE files that are not in Malsign, but we can query VirusTotal using the file hash. We consider that a file is malicious if at least 4 AV engines in the VT report had a detection label for it, a threshold also used in prior works to avoid false positives [35].

2.3 Problem Statement

In this paper we conduct a systematic analysis of PUP prevalence and its distribution through PPI services. We split our measurements in two main parts. First, we measure how prevalent PUP is. This includes what fraction of hosts have PUP installed, which are the top PUP publishers, and what is the installation base of PUP publishers in comparison with benign publishers. Then, we measure the PPI ecosystem including who are the top PPI services and PUP advertisers, what percentage of PUP installations are due to PPI services and advertiser affiliate programs, what are the relationships between PUP and malware, and what are the domains from where PUP is downloaded.

We do not attempt to differentiate what behaviors make a program PUP or malware, but instead rely on AV vendors for this. We leverage the prior finding that the majority of PUP (and very little malware) is properly signed. In particular, signed executables flagged by AV engines are predominantly PUP, while malware rarely obtains a valid code signing chain due to identity checks implemented by CAs [35]. Using that finding, we consider PUP any signed file flagged by at least 4 AV engines. Thus, the term PUP in this paper includes different types of files that AV vendors flag as PUP including undesirable advertiser programs, bundles of publisher programs with PPI installers, and stand-alone PPI installers.

To measure PUP prevalence, we first identify a list of dominant PUP publishers extracted from the code signing certificates from the 11M VT reports from the malware feed (Section 3). Then, we group publisher names (i.e., subject strings in code signing certificates) from the same entity into publisher clusters (Section 4). Finally, we use the WINE binary reputation data to measure the PUP installation base, as well as the installation base of individual PUP and benign publisher clusters (Section 5). Since we focus on signed executables, our numbers constitute a lower bound on PUP and publisher prevalence.

To measure the PPI ecosystem, we build a publisher graph that captures the who-installs-who relationships among PUP publishers. We use the graph for identifying PPI services and PUP advertisers (Section 6). Then, we measure the percentage of PUP installations due to PPI services and advertiser affiliate programs (Section 7). Next, we analyze the downloads of malware by PUP and the downloads of PUP by malware (Section 8). Finally, we examine the domains from where PUP is downloaded (Section 9).

3 Identifying PUP Publishers

The first step in our approach is to identify a list of dominant PUP publishers. As mentioned earlier, prior work has shown that signed executables flagged by AV engines are predominantly PUP, while malware is rarely properly signed. Motivated by this finding, we identify PUP publishers by ranking publishers of signed binaries flagged by AV vendors, by the number of samples they sign.

For this, we obtain a list of 11M potentially malicious samples from a "malware" feed and query them in VirusTotal to collect their VT reports. From these reports, we keep only executables flagged by at least 4 AV vendors to make sure we do not include benign samples in our study. We further filter out executables with invalid signatures, i.e., whose publisher information cannot be trusted. These filtering steps leave us with 2.5M binaries whose signatures validated at the time of signing. These include executables whose certificate chain still validates, those with a revoked certificate, and those with expired certificates issued by a valid CA.

From each of the 2.5M signed executables left, we extract the publisher's subject common name from the certificate information in its VT report. Hereinafter, we will refer to the publisher's subject common name as publisher name. Oftentimes, publisher names have some variations despite belonging to the same entity. For example, MyRealSoftware could use both "MyRealSoftware S.R.U" and "MyRealSoftware Inc" in the publisher name. Thus, we perform a normalization on the publisher names to remove company suffixes such as Inc., Ltd. This process outputs a list of 1,440 normalized PUP publisher names. Table 11 in the Appendix shows the top 20 normalized PUP publisher names by number of samples signed in the feed. These 20 publishers own 56% of the remaining signed samples after filtering.

Clearly, our list does not cover all PUP publishers in the wild. This would not be possible unless we analyzed all existing signed PUP. However, the fact that we analyze 2.5M of undesirable/malicious signed samples gives us confidence that we cover the top PUP publishers. Those 1,440 PUP publisher names are used to scan the file publisher field in WINE's binary downloads dataset to identify events that involve samples from those PUP publishers, i.e., where a parent or child file belongs to the 1,440 PUP publishers. As shown in Table 2, there are 8 B such events.

Note that at this point we still do not know whether different publisher names (i.e., entries in Table 11) belong to the same PUP publisher. For example, some popular publisher names such as Daniel Hareuveni, Stepan Rybin, and Stanislav Kabin are all part of Web Pick Internet Holdings Ltd, which runs the InstalleRex PPI service. The process to cluster publisher names that belong to the same publisher is described in Section 4.

4 Clustering Publishers

PUP authors use certificate polymorphism to evade detection by certification authorities and AV vendors [35]. Two common ways to introduce certificate polymorphism are applying small variations to reuse the same identity / publisher name (e.g. apps market ltd, APPS Market Inc., Apps market Incorporated) and using multiple identities (i.e., companies or persons) to obtain code signing certificates. We cluster publisher names that belong to the same publisher according to similarities on the publisher names, domain names in events with URLs, and Malsign clustering results.

Publisher name similarity. This feature aims to group together certificates used by the same identity that have small variations on the publisher name. Since the WINE binary downloads dataset contains the publisher name for parent and child files, this feature can be used even when a signed sample has no VT report and we do not have the executable (i.e., not in Malsign). The similarity between two publisher names is computed in two parts: first derive a list of normalized tokens from each publisher name through four steps and then compute similarity between the token lists.

To obtain the token list of a publisher name, the first step is to extract parenthesized strings as separate tokens. For example, given the publisher name "Start Playing (Start Playing (KnockApps Limited))" this step produces 3 tokens: "Start Playing", "Start Playing", and "KnockApps Limited". The second step converts each token to lowercase and removes all non-alphanumeric characters from the token. The third step removes from the tokens company extensions (e.g., ltd, limited, inc, corp), geographical locations (e.g., countries, cities), and the string

Publishers	Clusters	Singletons	Largest	Median
6,066	5,074	4,534	103	1

Table 3: Publisher clustering results.

"Open Source Developer", which appears in code signing certificates issued to individual developers of open source projects. Finally, tokens that have less than 3 characters and duplicate tokens are removed.

To compute the similarity between two token lists, for each pair of publisher names P_1 and P_2, we calculate the normalized edit distance among all token pairs (t_i, t_j) where t_i belongs to P_1 and t_j to P_2. If the edit distance between P_1 and P_2 is less than 0.1, we consider these two publishers to be the same. We selected this threshold after experimenting with different threshold values over 1,157 manually labeled publisher names. The edit distance threshold of 0.1 allowed us grouping the 1,157 publisher names into 216 clusters with 100% precision, 81.9% recall, and 86.4% F1 score.

Child download domains. If child executables signed by different publisher names are often downloaded from the same domains, that is a good indication that the publisher names belong to the same entity. To capture this behavior, we compute the set of ESLDs from where files signed by the same publisher name have been downloaded. Note that we exclude ESLDs that correspond to file lockers and download portals as they are typically used by many different publishers. The publisher names whose Jaccard Index of their ESLD sets is over 0.5 are put to the same cluster.

Parent download domains. Similarly, if parent files signed by different publisher names download from a similar set of domains, this indicates the publisher names likely belong to the same entity. This feature first computes the set of ESLDs from where parent files signed by the same publisher name download (excluding file lockers and download portals). Publisher names whose Jaccard Index is over 0.5 are put to the same cluster.

Malsign clustering. For each Malsign cluster we extract the list of distinct publisher names used to sign executables in the cluster, i.e., Subject CN strings extracted from certificates for files in the cluster. We consider that two publisher names in the same Malsign cluster belong to the same publisher.

Final clustering. We group publisher names into the same cluster if they satisfy at least one of the first 3 features explained above or are in the same Malsign cluster. Table 3 summarizes the clustering, which produces 5,074 clusters from 6,066 publisher names.

5 PUP Prevalence

In this Section, we measure the prevalence of PUP, based on the number of hosts in the WINE binary downloads dataset (i.e., WINE hosts) that have installed programs from PUP publishers. We measure the total number of WINE hosts affected by PUP, rank PUP publishers by installation base, and compare the installation base of PUP publishers to benign publishers.

We first compute the *detection ratio* (DR) for each cluster, which is the number of samples signed by publishers in the cluster flagged by at least 4 AVs, divided by the total number of samples in the cluster for which we have a VT report. We mark as PUP those clusters with $DR > 5\%$, a threshold chosen because is the lowest that leaves out known benign publishers. From this point on, when we refer to PUP publishers, we mean the 915 publisher clusters with $DR > 5\%$.

Note that the number of WINE hosts with installed programs from a publisher cluster constitutes a quite conservative lower bound on the number of hosts across the Internet that have programs installed from that publisher. It captures only those Symantec customers that have opted-in to share data and have been sampled into WINE. If we take into account that Symantec only had 8% of the AV market share in January 2014 [47] and that only $\frac{1}{16}$ of Symantec users that opt-in to share telemetry are sampled into WINE [6], we estimate that the number of WINE hosts is two orders of magnitude lower than the corresponding number of Internet-connected hosts. Furthermore, we do not count WINE hosts with only unsigned PUP executables installed.

PUP prevalence. We find 2.1M WINE hosts, out of a total 3.9M WINE hosts in our time period, with at least one executable installed from the 915 PUP clusters. Thus, 54% of WINE hosts have PUP installed. This ratio is a lower bound because we only count signed PUP executables (i.e., we ignore unsigned PUP executables) and also because our initial PUP publisher list in Section 3 may not be complete. Thus, PUP is prevalent: more than half of the hosts examined have some PUP installed.

Top PUP publishers. Table 4 shows the top 20 PUP publishers by WINE installation base and details the cluster name, whether the publisher is a PPI service (this classification is detailed in Section 6), the number of publisher names in the cluster, detection ratio, and host installation base. The number of publishers ranks from singleton clusters up to 48 publishers for IronSource, an Israeli PPI service. The installation bases for the top 20 PUP publishers range from 200K up to over 1M for Pe-

#	Cluster	PPI	Pub	DR	Hosts
1	Perion Network	✓	5	52%	1.0M
2	Mindspark	✗	1	85%	533K
3	Bandoo Media	✗	5	46%	373K
4	Web Pick	✓	21	79%	346K
5	IronSource	✓	48	81%	332K
6	Babylon	✗	1	38%	330K
7	JDI BACKUP	✗	1	56%	328K
8	Systweak	✗	3	37%	320K
9	OpenCandy	✓	1	55%	311K
10	Montiera Technologies	✗	2	54%	303K
11	Softonic International	✗	2	70%	292K
12	PriceGong Software	✗	1	18%	292K
13	Adknowledge	✓	7	75%	277K
14	Adsology	✗	2	77%	276K
15	Visual Tools	✗	2	70%	275K
16	BitTorrent	✗	1	40%	271K
17	Wajam	✗	2	87%	218K
18	W3i	✓	4	93%	216K
19	iBario	✓	15	84%	208K
20	Tuguu	✓	14	94%	200K

Table 4: Top 20 PUP publishers by installation base.

#	Cluster	PUP	Hosts
1	Microsoft	✗	3.9M
2	Symantec	✗	3.8M
3	Adobe Systems	✗	3.5M
4	Google	✗	3.1M
5	Apple	✗	1.8M
6	Intel	✗	1.6M
7	Sun Microsystems	✗	1.6M
8	Cyberlink	✗	1.6M
9	GEAR Software	✗	1.5M
10	Hewlett-Packard	✗	1.5M
11	Oracle	✗	1.4M
12	Skype Technologies	✗	1.3M
13	Mozilla Corporation	✗	1.0M
14	McAfee	✗	1.0M
15	Perion Network / Conduit	✓	1.0M
16	WildTangent	✗	941K
17	Macrovision Corporation	✗	802K
18	LEAD Technologies	✗	775K
19	NVIDIA Corporation	✗	722K
20	Ask.com	✗	624K
24	Mindspark Interactive Network	✓	533K

Table 5: Top publishers by install base (benign and PUP).

rion Network, an Israeli PPI service that bought the operations of the infamous Conduit toolbar in 2013. As explained earlier, these numbers are a quite conservative lower bound. We estimate the number of Internet-connected computers for these publishers to be two orders of magnitude larger, in the range of tens of millions, and up to a hundred million, hosts. We have found anecdotal information that fits these estimates. For example, an adware developer interviewed in 2009 claimed to have control over 4M machines [12].

Comparison with benign publishers. Table 5 shows the top 20 publisher clusters, benign and PUP, in WINE. The most common publishers are Microsoft and Symantec that are installed in nearly all hosts. The Perion Network / Conduit PPI network ranks 15 overall. That is, there are only 14 benign software publishers with a larger installation base than the top PUP publisher. Perion Network is more prevalent than well known publishers such as Macrovision and NVIDIA. The second PUP publisher (Mindspark Interactive Network) has the rank 24. This highlights that top PUP publishers are among the most widely installed software publishers.

A reader may wonder if we could also compute the installation base for malware families. Unfortunately, due to malware being largely unsigned and highly polymorphic, we would need to first classify millions of files in WINE (without having access to the binaries) before we can perform the ranking.

6 Classifying Publishers

Among the 5,074 PUP publisher clusters obtained in Section 4 we want to identify important clusters playing a specific role in the ecosystem. In particular, we want to identify clusters that correspond to PPI services and to examine the type of programs distributed by the dominant advertisers. For this, we first build a *publisher graph* that captures the who-installs-who relationships. Then, we apply filtering heuristics on the publisher graph to select a subset of publishers that likely hold a specific role, e.g., PPI service. Finally, we manually classify the filtered publishers into roles by examining Internet resources, e.g., publisher web pages, PPI forums, and the Internet Archive [32].

Publisher graph. The publisher graph is a directed graph where each publisher cluster is a node and an edge from cluster C_A to cluster C_B means there is at least one event where a parent file from C_A installed a child file from C_B. Self-edges are excluded, as those indicate program updates and downloads of additional components from the same publisher. Note that an edge captures download events between parent and child clusters across all hosts and the 19 months analyzed. Thus, the publisher graph captures the who-installs-who relationships over that time period, enabling a birds-eye view of the ecosystem.

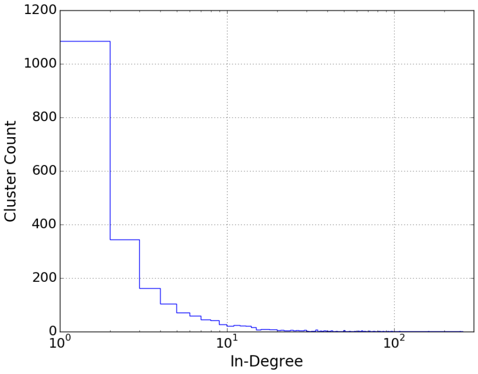

Figure 2: Cluster in-degree distribution.

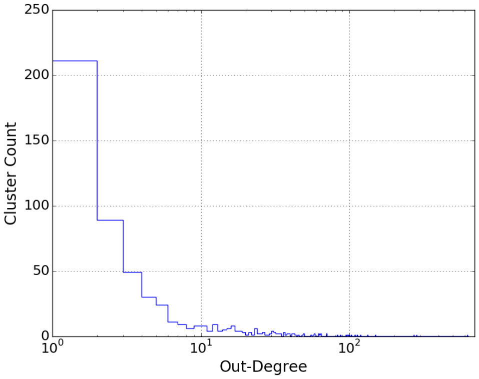

Figure 3: Cluster out-degree distribution.

In-degree and out-degree. We first measure the in-degree and out-degree of each cluster in the publisher graph. The in-degree is the count of distinct parent publisher clusters that install programs from a child publisher cluster. Intuitively, publishers with a high in-degree are installed by many other publishers, which indicates that they are buying installs. The out-degree is the count of distinct child publisher clusters installed by a parent publisher cluster. Intuitively, publishers with a high out-degree install many other publishers, which indicates that they are selling installs.

To compute a cluster's in-degree we filter out 12 benign parent clusters that correspond to tools that download large numbers of executables from different publishers such as browsers, BitTorrent clients, and Dropbox. To compute a cluster's out-degree we exclude benign child publishers ($DR < 5\%$) that are typically dependencies.

Figure 2 shows the in-degree distribution. 57% of the clusters have no parents (i.e., installed by unsigned files only). Another 21.5% have one. These are typically installed only by parents in the same cluster. Only 224 (4.4%) clusters have an in-degree larger than 10. We call these high in-degree clusters. Figure 3 shows the out-degree distribution. 572 clusters (11%) have an out-degree larger than zero and only 133 (2.6%) clusters have an out-degree larger than 10. We call these high out-degree clusters.

PPI services. To identify PPI services in the publisher graph, we first select all PUP publisher clusters with both high in-degree and high out-degree (i.e., $DR \geq 5\% \wedge ID \geq 10 \wedge OD \geq 10$), which indicate these publishers are buying and selling installs. This rule reduces the 5,074 clusters to 49 candidate publisher clusters. Next, we manually classify those 49 clusters through extensive analysis using PPI forums, publisher websites, and the Internet Archive. Of those 49 clusters, we classify 22 as PPI services, 12 as advertisers that run an affiliate program, 8 as advertisers without an affiliate program; 3 as download portals (Download.com, BrotherSoft, Softonic), and 4 as PUP publishers that distribute free download tools (e.g., BitTorrent clients). The latter tools inflate the out-degree of their publishers and were not included in our whitelist of download tools due to the high DR of their publishers. Our manual analysis also reveals two additional PPI services (7install and Install Monster) that were missed by our rule because they do not achieve high enough in-degree and out-degree, either because of low popularity or because they appear at the end of our observation period.

Table 6 summarizes the 24 identified PPI services sorted by installation base. For each cluster, it shows the name of the PPI service, in-degree, out-degree, installation base, detection ratio, and number of publishers in the cluster. The classification reveals that 3 of the top 5 PUP publishers by installation base in Table 4 are PPI services. Thus, the most popular PUP publishers are PPI services. Some of the PPI services identified no longer work at the time this paper is published, e.g., OneInstaller, but their PPI service front-ends are present in the Internet Archive.

During our manual analysis we keep track of all PPI services we find advertised on the Internet, e.g., on PPI forums. In addition to the 24 PPI services in Table 6 we identify another 12 PPI services, shown in Table 12 in the Appendix. There are several reasons for which we do not observe those 12 PPI services in our data. First, some of them are simply resellers that pay affiliate publishers to distribute bundles or downloaders for other PPI services. Second, PPI services may have been launched

#	Cluster	PPI Service	ID	OD	Hosts	DR	Pub.
1	Perion Network/Conduit	CodeFuel [10]	168	63	1 M	52%	5
2	Yontoo	Sterkly [63]	53	17	601 K	93%	103
3	iBario	RevenueHits [56]	62	36	479 K	84%	16
4	Web Pick	InstalleRex [27]	65	22	346 K	79%	21
5	IronSource	InstallCore [26]	73	112	332 K	81%	48
6	OpenCandy	OpenCandy [46]	91	36	311 K	55%	1
7	Adknowledge	Adknowledge [20]	53	48	277 K	75%	7
8	W3i	NativeX [41]	38	49	216 K	93%	4
9	Somoto	BetterInstaller [5]	60	70	209 K	96%	5
10	Firseria	Solimba [62]	41	30	209 K	94%	9
11	Tuguu	DomaIQ [13]	49	16	200 K	94%	14
12	Download Admin	DownloadAdmin [14]	25	16	192 K	73%	2
13	Air Software	AirInstaller [3]	33	41	191 K	79%	1
14	Vittalia Internet	OneInstaller [45]	27	29	155 K	71%	18
15	Amonetize	installPath [31]	50	63	154 K	93%	2
16	SIEN	Installbay [25]	34	33	139 K	80%	2
17	OutBrowse	RevenYou [57]	22	41	86 K	94%	4
18	Verti Technology Group	Verti [66]	17	39	47 K	44%	1
19	Blisbury	Smart WebAds [60]	19	30	46 K	77%	2
20	Nosibay	Nosibay [44]	19	20	30 K	75%	1
21	ConversionAds	ConversionAds [11]	10	38	24 K	72%	1
22	Installer Technology	InstallerTech [28]	10	14	11 K	56%	1
23	7install	7install [1]	2	0	75	12%	1
24	Install Monster	Install Monster [30]	3	1	9	100%	1

Table 6: PPI services services identified sorted by installation base.

(or gained popularity) after the end of our observation period (e.g., AdGazelle). Third, some PPI services may distribute unsigned bundles or downloaders. For example, we examined over 30K samples that AV engines label as belonging to the InstallMonetizer PPI service, of which only 8% were signed. Finally, some PPI services may have so low volume that they were not observed in our initial 11 M sample feed.

Advertisers. To identify advertisers in the publisher graph, we first select PUP clusters with high in-degree, low out-degree, and for which at least one parent is one of the 24 PPI services (i.e., $DR \geq 5\% \wedge ID \geq 10 \wedge OD \leq 9 \wedge PPPI > 0$). Advertisers pay to have their products installed (i.e., buy installs) and may not install other publishers for monetization as they know how to monetize the machines themselves. Since buying installs costs money, they need to generate enough income from the installations to offset that cost. This filtering identifies 77 clusters, which we manually examine to identify the main product they advertise (they can advertise multiple ones) and whether they run an affiliate program where they pay affiliates to distribute their programs. We also include in this analysis the 20 advertiser clusters manually identified in the PPI service identification above.

Table 7 shows the top 30 advertiser clusters by installation base. The table shows the cluster name, whether it runs an affiliate program, in-degree, out-degree, detection ratio, installation base, the number of parent PPI service nodes, the number of child PPI service nodes, the main product they install, and whether they install browser add-ons (BAO). The latter includes any type of browser add-ons such as toolbars, extensions, plugins, browser helper objects, and sidebars.

The data shows that 18 of the 30 top advertisers install browser add-ons. Those browser add-ons enable monetization through Web traffic, predominantly through different types of advertisement. Common methods are modifying default search engines to monetize searches (e.g., SearchResults, Delta Toolbar, Imminent Toolbar), shopping deals and price comparisons (e.g., PriceGong, PricePeep, DealPLY, SupremeSavings), and other types of advertisement such as pay-per-impression and pay-per-action (e.g., Widgi Toolbar, Inbox Toolbar).

The 12 advertisers that focus on client applications monetize predominantly through selling licenses and subscriptions. The main group is 6 publishers advertising rogueware claiming to improve system performance (Regclean Pro, Optimizer Pro, SpeedUpMyPC,

#	Cluster	Aff	ID	OD	DR	Hosts	PPPI	CPPI	Main Product	BAO
1	Xacti	✗	57	9	22%	563 K	13	1	RebateInformer	✓
2	Mindspark	✓	62	17	85%	533 K	3	5	Mindspark Toolbar	✓
3	Bandoo Media	✓	86	108	46%	373 K	7	18	MediaBar	✓
4	Babylon	✓	83	14	38%	330 K	16	3	Babylon Toolbar	✓
5	JDI Backup Limited	✓	71	19	56%	328 K	17	3	MyPC Backup	✗
6	Systweak	✓	81	24	37%	320 K	7	2	Regclean Pro	✗
7	Montiera Technologies	✗	37	2	66%	303 K	8	1	Delta Toolbar	✓
8	PriceGong Software	✗	12	0	17%	292 K	6	0	PriceGong	✓
9	Adsology	✓	62	12	77%	276 K	17	1	OptimizerPro	✗
10	Wajam	✗	42	5	87%	218 K	11	2	Wajam	✓
11	Visicom Media	✗	13	2	14%	185 K	4	0	VMN Toolbar	✓
12	Linkury	✗	46	2	54%	174 K	13	0	SmartBar	✓
13	Uniblue Systems	✓	64	13	11%	160 K	10	1	SpeedUpMyPC	✗
14	Search Results	✗	35	3	79%	159 K	12	2	SearchResults	✓
15	Bitberry Software	✓	13	64	88%	130 K	1	7	BitZipper	✗
16	Iminent	✗	13	1	74%	118 K	4	1	Iminent Toolbar	✓
17	DealPly Technologies	✗	43	0	93%	108 K	16	0	DealPly	✓
18	Smart PC Solutions	✓	38	0	32%	106 K	13	0	PC Speed Maximizer	✗
19	DVDVideoSoft	✗	15	2	18%	101 K	3	1	Free Studio	✗
20	Spigot	✓	17	1	39%	101 K	1	1	Widgi Toolbar	✓
21	Web Cake	✗	34	2	98%	97 K	16	2	Desktop OS	✓
22	GreTech	✓	13	1	21%	90 K	3	1	GOM Player	✗
23	Digital River	✓	17	0	10%	80 K	1	0	DR Download Manager	✓
24	Widdit	✓	20	16	27%	79 K	4	2	HomeTab	✓
25	EpicPlay	✗	12	4	90%	77 K	3	1	EpicPlay	✗
26	Iobit Information Technology	✓	18	8	6%	73 K	3	1	Advanced SystemCare	✗
27	DT Soft	✗	14	2	22%	68 K	2	1	DAEMON Tools	✗
28	Innovative Apps	✗	14	1	68%	60 K	7	0	Supreme Savings	✓
29	Woolik Technologies	✗	13	9	70%	50 K	4	1	Woolik Search Tool	✓
30	Visual Software Systems	✓	22	12	62%	42 K	5	3	VisualBee	✗

Table 7: Top 30 advertiser clusters by installation base. For each publisher cluster it shows: whether we found an affiliate program (Aff), the in-degree (IN), out-degree (OD), detection ratio (DR), installation base (Hosts), number of parent PPI services (PPPI), number of child PPI services (CPPI), the main product advertised, and whether that product is a browser add-on (BAO) including toolbars, extensions, sidebars, and browser helper objects.

Event Type	Count
All PUP downloads	40.1M
Unsigned parent	11.5M
Signed parent	28.6M
Benign parent	7.4M
PUP parent	21.2M
PPI	7.3M
Adv. affiliate program	5.5M

Table 8: Analysis of PUP download events.

PC Speed Maximizer, Advanced System Care, DAEMON Tools). These rogueware try to convince users to buy the license for the full version. We also observe multimedia tools (Free Studio, GOM Player), backup tools (MyPC Backup), game promotion (EpicPlay), compressors (BitZipper), and presentation tools (Visual Bee).

7 PUP Distribution Methods.

This section measures the distribution of PUP through PPI services and affiliate programs. The relevant data is provided in Table 8. From the 90 M events with parent information, we first find the events with child files that are signed by PUP publishers (40.1M events). Then, we investigate the parents that installed them. In 28.6M (71%) of these events, parents were signed, therefore allowing us to go further in our search for finding the parents who are PPIs. 7.4M (35%) of the these parents correspond to Web browsers and other benign download programs such as BitTorrent clients and Dropbox. The remaining 21.2M (65%) events have a PUP parent. This indicates that the majority of PUP is installed by other PUP. In particular, for 7.3M out of 21.2M events (34%)

with PUP parent, the parent corresponds to one of the 24 PPI services identified in Table 6. And, for another 5.5M (26%) events the parent corresponds to one of the 21 affiliate programs identified in Section 6. From these statistics, we can conclude that PUPs are generally installed by other PUPs and moreover, over 25% of the PUP download events are sourced by PPI services, and another 19% by advertisers with affiliate programs.

8 PUP–Malware Relationships

We are interested in understanding if there is any form of relationship between PUP and malware and if malware uses the PPI services we identified. In particular we would like to measure the percentage of PUP that installs malware or is installed by malware. Here, the obvious challenge is to accurately label malware in the WINE dataset. While the majority of properly signed executables flagged by AV engines are PUP, unsigned executables flagged by AV engines can be PUP or malware and there are a few malware that are signed.

To address these issues, we use AVClass, a recently released malware labeling tool [58]. Given the VT reports of a large number of executables, AVClass addresses the most important challenges in extracting malware family information from AV labels: label normalization, generic token detection, and alias detection. For each sample, it outputs a ranking of the most likely family names ranked by the number of AV engines assigning that family to the sample. Since AV labels can be noisy [4], we focus on executables for which the top family AVClass outputs is in a precomputed list of 70 malware families that includes prevalent families such as zbot, zeroaccess, reveton, virut, sality, shylock, and vobfus. Clearly, our methodology is not 100% accurate, but allows us to gain insight on the relationships between malware and PUP.

PUP downloading malware. One way malware authors could relate to PUP could be by signing up as advertisers to PPI services to distribute their malware. To identify such cases, we look for PUP publishers that download executables from one of the 70 malware families considered. What we have found out is that there is a link between 71 of the PUP publisher clusters to malware. Those publishers distribute malware from 40 families through 5,586 download events. Out of those 71 clusters, 11 are classified as PPI services in Section 6. Those PPI services generate 35% of the 5,586 malware downloads by PUP. For example, Perion Network, the most popular PPI service, downloads instances of zbot, shylock, and andromeda trojans. We also observe at the end of 2013 iBario downloading instances of sefnit click-fraud malware as reported by TrendMicro [38]. Clearly, 5,586 downloads is a low number, which may indicate that malware favors silent distribution vectors and that PPI services are careful to avoid malware to preserve their reputation towards security vendors. We only observe occasional events spread amongst multiple PPI services, possibly due to insufficient checks by those PPI services. Another factor of influence may be that installs through these PPIs can be an order of magnitude more expensive than those from silent PPIs, as shown in Section 2.1.

Malware downloading PUP. Malware authors could also sign up as affiliate publishers to PPI services to monetize the compromised machines by selling installs. To capture this behavior, we analyzed PUP downloaded by samples from the 70 malware families considered. We found 11K downloads by malware from 25 families. These malware samples downloaded executables from 98 PUP publisher clusters. 88% of these downloads were generated by 3 malware families: vobfus, badur, and delf. 7 of the 98 PUP publisher clusters belong to the PPI services category. For example, we observe zeroaccess installing files from the DomaIQ PPI service. Overall, malware downloading PUP is a more common event than PUP downloading malware, but still rare, affecting only 0.03% of all events where PUP is downloaded.

The conclusion of this analysis is that while PUP–malware interactions exist, they are not prevalent and malware distribution seems disjoint from PUP distribution. Observed malware–PPI service interactions do not focus on a few misbehaving PPI services, but rather seem to occasionally affect many PPI services.

9 Domain Analysis

In this section we analyze the 1.1 M events that contain a URL, and in particular the domains (ESLDs) in those URLs. The events that contain a URL allow us to identify publishers that download from and are downloaded from a domain. Note that the domains we extract from this dataset are used for hosting and distributing executables and do not cover all of the domains used by PUP. We identify 3 main types of domains from our analysis:

- **File lockers.** Cloud storage services used for backup or sharing executables between users. They exhibit a high number of client publishers being downloaded from them, most of which are benign (e.g., Microsoft, Adobe, AutoDesk). These ESLDs also host a front-end website for users.

- **Download portals.** They also distribute programs from a high number of publishers, predominantly free software publishers and their own PPI services. They also host a front-end website.

- **PPI services.** Used by PPI services to host their wrappers and advertised programs. These ESLDs do not host a front-end website as they are accessed by PPI installers, rather than humans.

Rank by downloaded publishers. Table 9 shows the top 20 ESLDs by number of child publishers signing files downloaded from that ESLD. The 4 tick-mark columns classify the domain as file locker (FL), download portal (DP), PPI service (PPI), or other (Oth). Of the 20 ESLDs, 15 correspond to file lockers, 2 to download portals, and another 2 to PPI services. The remaining domain is `file.org`, a portal where users can enter a file extension to find a tool that can open files with that extension. The publisher behind this portal uses it to promote its own free file viewer tool, which is offered as the best tool to handle over 200 file extensions.

If we give a vote to the top 3 publishers downloaded from each of the 15 file lockers (45 votes), Microsoft gets 13, Adobe 11, Cyberlink 4, and AutoDesk 3. The rest are popular benign publishers such as Ubisoft, VMWare, and Electronic Arts. Thus, file lockers predominantly distribute software from reputable publishers.

For the two download portals, the publishers downloaded from them correspond to their own PPI service (i.e., bundles signed by "CBS Interactive" from cnet.com), free software publishers, and PPI services. For `edgecastcdn.net` all 67 publishers are part of the same PPI service run by the Yontoo group. The domain `d3d6wi7c7pa6m0.cloudfront.net` belongs to the Adknowledge PPI service and distributes their advertiser programs. Among those advertiser programs we observe bundles signed by other PPI services, which may indicate arbitrageurs who try to take advantage of pricing differentials among PPI services [7].

Rank by downloads. Table 10 ranks the top 20 domains by number of downloads. It shows the ESLD, the type (file locker, download portal, PPI service, advertiser, other), the cluster that owns the domain, the number of downloads, the number of publishers of the downloaded executables, and the number of distinct files downloaded. We label each domain as belonging to the cluster that signs most executables downloaded from the domain. The publisher in the other category is Frostwire, which distributes a popular free BitTorrent client.

ESLD	FL	DP	PPI	Oth	Pub
uploaded.net	✓				366
cnet.com		✓			142
extabit.com	✓				128
share-online.biz	✓				125
4shared.com	✓				120
rapidgator.net	✓				90
depositfiles.com	✓				76
mediafire.com	✓				73
edgecastcdn.net			✓		67
chip.de		✓			53
zippyshare.com	✓				49
uloz.to	✓				48
file.org				✓	47
putlocker.com	✓				47
d3d6wi7c7pa6m0.cf			✓		44
turbobit.net	✓				44
freakshare.com	✓				41
rapidshare.com	✓				40
ddlstorage.com	✓				38
bitshare.com	✓				38

Table 9: Top 20 ESLDs by number of distinct publishers of downloaded executables. FL means file locker, DP download portal, PPI pay-per-install service, and Oth other. For brevity, d3d6wi7c7pa6m0.cf stands for d3d6wi7c7pa6m0.cloudfront.net.

Table 10 shows that PPI domains dominate in terms of downloads, but distribute a smaller number of child publishers compared to file lockers and download portals that dominate Table 9. It also shows that it is possible to link download domains to the publishers that own them based on the signature of files they distribute, despite the domains being typically registered by privacy protection services.

10 Discussion

Unsigned PUP. Our work focuses on signed PUP executables based on the prior observation that most signed samples flagged by AV engines are PUP [35]. However, this means that we will miss PUP publishers if they distribute only unsigned executables. Also, our PUP prevalence measurements are only a lower bound since there may be hosts with only unsigned PUP installed. In concurrent work, Thomas et al. [65] infiltrate 4 PPI services observing that only 58% of the advertiser software they distribute is signed. Thus, we could be missing as much as 42% of PUP software, but we expect a much smaller number of hosts will only have unsigned PUP installed.

ESLD	FL	DP	PPI	Ad	Oth	Cluster	Downl.	Pub.	Children
conduit.com			✓			Perion Network	138,480	2	727
edgecastcdn.net			✓			Yontoo	106,449	67	1,148
frostwire.com					✓	Frostwire	53,592	1	2,511
ask.com				✓		Ask	40,939	6	125
imgfarm.com				✓		Mindspark	26,498	6	3,209
ilivid.com				✓		Bandoo Media	25,429	5	905
conduit-services.com			✓			Perion Network	21,149	8	1,345
adpk.s3.amazonaws.com				✓		Adpeak	14,513	2	36
airdwnlds.com			✓			Air Software	14,342	1	13,389
ncapponline.info			✓			Web Pick	13,974	11	13,252
uploaded.net	✓					Cyando	10,886	366	7,816
storebox1.info			✓			Web Pick	10,109	13	9,561
oi-installer9.com			✓			Adknowledge	8,360	4	7,892
4shared.com	✓					4shared	8,222	120	5,649
systweak.com				✓		Systweak	8,104	4	509
mypcbackup.com				✓		JDI Backup Limited	7,837	1	43
greatfilesarey.asia			✓			Web Pick	7,699	8	7,296
incredimail.com			✓			Perion Network	7,408	3	2,571
softonic.com		✓				Softonic	6,980	36	3,869
nicdls.com			✓			Tuguu	6,908	14	1,704

Table 10: Top ESLDs by number of downloads from them. The two rightmost columns are the number of publishers and files of the downloads.

Affiliate publisher analysis. We have classified publisher clusters as PPI services and advertisers, but we have not examined affiliate publisher clusters. One challenge with affiliate publishers is that when distribution happens through a stand-alone PPI installer (rather than bundles) both the advertiser program and the affiliate publisher program may appear as children of the PPI service in the publisher graph. It may be possible to measure the number of affiliates for some PPI services by analyzing URL parameters of download events. We leave this analysis to future work.

Other distribution models. We have examined PUP distribution through PPI services and advertiser affiliate programs. However, other distribution models exist. These include bilateral distribution agreements between two parties (e.g., Oracle's Java distributing the Ask toolbar [34]) and pre-installed PUP (e.g., Superfish on Lenovo computers [21]). We observe Superfish distributed through PPI services prior to the Lenovo agreement, which started in September 2014 after our analysis period had ended. We leave the analysis of such distribution models to future work.

Observation period. Our observation period covers 19 months from January 2013 to July 2014. Unfortunately, WINE did not include newer data at the time of our study. Thus, we miss newer PUP publishers that joined the ecosystem after our observation period. However, the vast majority of PUP publishers examined are still alive at the time of writing.

Internet population. We have measured the installation base of PUP (and benign) publishers on WINE hosts. We have also estimated that our measured WINE population may be two orders of magnitude lower than that of hosts connected to the Internet. But, we concede that this estimation is rough and could be affected by different factors such as selection bias.

11 Related Work

PUP. Potentially unwanted programs have received little attention from academia. In 2005–2007 Edelman studied the deceptive installation methods by spyware and other unwanted software [19]. In 2012, Pickard and Miladinov [52] studied a PUP rogue anti-malware software concluding that while not malicious, it only detected 0.3% of the malware and its main purpose was convincing the user to pay the license. Recently, some works have hinted at the increased prevalence and importance of PUP. Thomas et al. [64] study ad injectors, a type of PUP that modifies browser sessions to inject advertisements, finding that 5% of unique daily IP addresses accessing Google are impacted. In follow up work, Jagpal et al. [33] design WebEval, a system to identify malicious extensions at the core of ad injection. Kotzias et al. [35] analyze abuse in Windows Authenticode by analyzing 356K samples from malware feeds. They find that PUP has been quickly increasing feeds since 2010, that the vast majority of properly signed samples are PUP, and that PUP publishers use high file and certificate polymorphism to evade security tools and CA defenses such as identity validation and revocation.

In concurrent work, Thomas et al. [65] analyze the advertiser software distributed to US hosts by 4 PPI services (OutBrowse, Amonetize, OpenCandy, InstallMonetizer). They also use SafeBrowsing data to measure that PPI services drive over 60 million download events every week, nearly three times that of malware. Both works are complementary in their study of PPI services and measuring users affected by PUP. They use a top-to-bottom approach of infiltrating a few PPI services plus SafeBrowsing data, while we perform a bottom-to-top approach starting from files installed on end hosts. We analyze 19 months from January 2013 to July 2014, while they analyze 12 months from August 2015 to July 2016. By examining download events on 3.9M WINE hosts in different countries, our approach enables us to measure PUP prevalence and achieves a broader coverage of the PPI ecosystem. We observe 23 PPI services including 3 of the 4 in their study. The missing PPI service is InstallMonetizer, which distributes mostly unsigned installers.

Also in concurrent work, Nelms et al. [42] analyzed web-based social engineering attacks that use deceiving advertisements to convince users to download unwanted software. They find that most programs distributed this way are bundles of free software with PUP.

Malware distribution. Prior work has studied malware distribution through different vectors, which differs from our focus on PUP distribution. Moschuk et al. [40] crawl 18M URLs finding that 5.9% were drive-by downloads and 13.4% lead to spyware. Provos et al. [53] study the prevalence of distribution through drive-by downloads. Grier et al. [22] analyze the commoditization of drive-by downloads and compare malware distribution through different vectors, concluding that drive-by downloads dominate. Caballero et al. [7] study malware distribution through PPI services. The PPI services we study differ in that installations are not silent and are mostly used by PUP and benign software. Kwon et al. [36] recently use WINE data to investigate malware distribution through downloaders. Their work differs in that they do not distinguish malware from PUP and in that they analyze file download graphs for individual machines. Instead, we analyze download relationships between publishers on aggregate over 3.9M machines over a 19 month time period, focusing on PUP distribution through PPI services and affiliate programs.

12 Conclusion

We have performed the first systematic study of PUP prevalence and its distribution through PPI services. By using AV telemetry comprising of 8 billion events on 3.9 million hosts over 19 months, we have found that over half (54%) of the examined hosts have PUP installed. The top PUP publishers are highly popular; the top PUP publisher ranks 15 amongst all software publishers (benign or not). We have built the publisher graph that captures the who-installs-who relationships between PUP publishers. We have identified that 65% of the PUP is installed by other PUP and that 24 PPI services distribute over 25% of the PUP and advertiser affiliate programs an additional 19%. We have examined the PUP-malware relationships finding 11K events where popular malware families install PUP for monetization and 5,586 events where PUP distributes malware. PUP-malware interactions are not prevalent and seem to occasionally affect most top PPI services. We conclude that PUP distribution is largely disjoint from malware distribution.

13 Acknowledgments

We thank Richard Rivera for his help with the clustering. This research was partially supported by the Regional Government of Madrid through the N-GREENS Software-CM project S2013/ICE-2731 and by the Spanish Government through the Dedetis Grant TIN2015-7013-R. All opinions, findings and conclusions, or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] 7install. https://web.archive.org/web/20160306081435/http://7install.com/.

[2] AdGazelle. http://adgazelle.com/.

[3] AirSoftware. https://airinstaller.com/.

[4] M. Bailey, J. Oberheide, J. Andersen, Z. M. Mao, F. Jahanian, and J. Nazario. Automated Classification And Analysis Of Internet Malware. In *International Symposium on Recent Advances in Intrusion Detection*, Queensland, Australia, September 2007.

[5] BetterInstaller. http://betterinstaller.somotoinc.com/.

[6] L. Bilge and T. Dumitras. Before We Knew It: An Empirical Study of Zero-day Attacks in the Real World. In *ACM Conference on Computer and Communications Security*, 2012.

[7] J. Caballero, C. Grier, C. Kreibich, and V. Paxson. Measuring pay-per-install: The commoditization of malware distribution. In *USENIX Security*, 2011.

[8] CashMyLinks. http://www.cashmylinks.com/.

[9] Cinstaller. http://cinstaller.com/.

[10] CodeFuel. 7 reasons codefuel beats all other pay per install companies, 2015. http://www.codefuel.com/blog/7-reasons-perion-codefuel-beats-all-other-pay-per-install-companies/.

[11] ConversionAds. https://web.archive.org/web/20160217095842/http://www.conversionads.com/.

[12] S. Davidoff. Interview with an adware author, 2009. http://philosecurity.org/2009/01/12/interview-with-an-adware-author.

[13] DomaIQ. http://www.domaiq.com/en/.

[14] Download Admin. https://web.archive.org/web/20140208040640/http://www.downloadadmin.com/.

[15] Download.com. http://www.download.com/.

[16] T. Dumitraş and D. Shou. Toward a Standard Benchmark for Computer Security Research: The Worldwide Intelligence Network Environment (WINE). In *EuroSys Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, April 2011.

[17] T. Dumitras and P. Efstathopoulos. The Provenance Of Wine. In *European Dependable Computing Conference*, May 2012.

[18] EarnPerInstall. https://web.archive.org/web/20160419013909/http://www.earnperinstall.com/.

[19] B. Edelman. Spyware Installation Methods. http://www.benedelman.org/spyware/installations/.

[20] M. Geary. Adknowledge apps distribution opportunities, 2013. http://ppitalk.com/showthread.php/49-Adknowledge-Apps-Distribution-Opportunities.

[21] D. Goodin. Lenovo pcs ship with man-in-the-middle adware that breaks https connections, 2015. http://arstechnica.com/security/2015/02/lenovo-pcs-ship-with-man-in-the-middle-adware-that-breaks-https-connections/.

[22] Grier et al. Manufacturing Compromise: The Emergence Of Exploit-as-a-service. In *ACM Conference on Computer and Communications Security*, Raleigh, NC, October 2012.

[23] GuppyGo. http://www.guppygo.com/.

[24] Installaxy. https://web.archive.org/web/20151105011933/http://installaxy.com/.

[25] InstallBay. http://www.visibay.com/installbay.

[26] InstallCore. https://www.installcore.com/.

[27] InstalleRex. https://installerex.com/.

[28] Installertech. http://www.installertech.com/.

[29] InstallMonetizer. http://www.installmonetizer.com/.

[30] InstallMonster. http://installmonster.ru/en.

[31] installPath. http://www.installpath.com.

[32] Internet Archive WayBack Machine. https://archive.org/web/.

[33] N. Jagpal, E. Dingle, J.-P. Gravel, P. Mavrommatis, N. Provos, M. A. Rajab, and K. Thomas. Trends and Lessons from Three Years Fighting Malicious Extensions. In *USENIX Security Symposium*, 2015.

[34] O. Java. What are the Ask Toolbars? https://www.java.com/en/download/faq/ask_toolbar.xml.

[35] P. Kotzias, S. Matic, R. Rivera, and J. Caballero. Certified PUP: Abuse in Authenticode Code Signing. In *ACM Conference on Computer and Communication Security*, 2015.

[36] B. J. Kwon, J. Mondal, J. Jang, L. Bilge, and T. Dumitras. The Dropper Effect: Insights into Malware Distribution with Downloader Graph Analytics. In *ACM SIGSAC Conference on Computer and Communications Security*, 2015.

[37] Mediakings. https://web.archive.org/web/20140517213640/http://media-kings.com/.

[38] T. Micro. On the actors behind mevade/sefnit, 2014. http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-on-the-actors-behind-mevade-sefnit.pdf.

[39] Microsoft. Windows authenticode portable executable signature format, Mar. 21 2008. http://download.microsoft.com/download/9/c/5/9c5b2167-8017-4bae-9fde-d599bac8184a/Authenticode_PE.docx.

[40] A. Moschuk, T. Bragin, S. D. Gribble, and H. Levy. A Crawler-based Study of Spyware in the Web. In *Network and Distributed System Security Symposium*, San Diego, CA, 2006.

[41] NativeX. http://nativex.com/.

[42] T. Nelms, R. Perdisci, M. Antonakakis, and M. Ahamad. Towards Measuring and Mitigating Social Engineering Malware Download Attacks. In *USENIX Security Symposium*, August 2016.

[43] Net Cash Revenue. http://netcashrevenue.com/.

[44] Nosibay. http://www.nosibay.com/.

[45] Oneinstaller. https://web.archive.org/web/20150220020855/http://oneinstaller.com/.

[46] Open Candy. http://opencandy.com/.

[47] Opswat Antivirus and Threat Report, January 2014. https://www.opswat.com/resources/reports/antivirus-january-2014.org/.

[48] PayPerInstall. http://payperinstall.com/.

[49] Perinstallbox. http://www.setupbundle.com/index.php.

[50] PerInstallBucks. https://perinstallbucks.com/.

[51] PerInstallCash. http://www.perinstallcash.com/.

[52] C. Pickard and S. Miladinov. Rogue software: Protection against potentially unwanted applications. In *Malicious and Unwanted Software (MALWARE), 2012 7th International Conference on*, pages 1–8. IEEE, 2012.

[53] N. Provos, P. Mavrommatis, M. A. Rajab, and F. Monrose. All Your Iframes Point To Us. In *USENIX Security Symposium*, San Jose, CA, July 2008.

[54] Public Suffix List. https://publicsuffix.org/.

[55] Purebits. http://purebits.net/.

[56] RevenueHits. https://web.archive.org/web/20130805140617/http://www.revenuehits.com/.

[57] RevenYou. http://www.revenyou.com/.

[58] M. Sebastián, R. Rivera, P. Kotzias, and J. Caballero. AVClass: A Tool for Massive Malware Labeling. In *International Symposium on Research in Attacks, Intrusions and Defenses*, September 2016.

[59] P. Security. Malware still generated at a rate of 160,000 new samples a day in Q2 2014. http://www.pandasecurity.com/mediacenter/press-releases/malware-still-generated-rate-160000-new-samples-day-q2-2014/.

[60] Smart WebAds. http://www.smartwebads.com/.

[61] Softonic. www.softonic.com.

[62] Solimba. https://solimba.com/.

[63] Sterkly. http://www.sterkly.com/.

[64] K. Thomas, E. Bursztein, C. Grier, G. Ho, N. Jagpal, A. Kapravelos, D. McCoy, A. Nappa, V. Paxson, P. Pearce, N. Provos, and M. A. Rajab. Ad Injection at Scale: Assessing Deceptive Advertisement Modifications. In *IEEE Symposium on Security and Privacy*, May 2015.

[65] K. Thomass, J. A. E. Crespo, R. Rastil, J.-M. Picodi, L. Ballard, M. A. Rajab, N. Provos, E. Bursztein, and D. Mccoy. Investigating Commercial Pay-Per-Install and the Distribution of Unwanted Software. In *USENIX Security Symposium*, Aug. 2016.

[66] Verti. http://www.vertitechnologygroup.com.

[67] VirusTotal. http://www.virustotal.com/.

[68] G. Wicherski. pehash: A novel approach to fast malware clustering. In *2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2009.

A Additional Results

Rank	Publisher	Samples	
1	Popeler System	326,530	13.2%
2	Daniel Hareuveni	138,159	5.6%
3	Start Now	117,930	4.8%
4	Mail.Ru	117,920	4.8%
5	Softonic International	69,233	2.8%
6	Bon Don Jov	68,937	2.8%
7	Stepan Rybin	68,390	2.8%
8	WeDownload	66,332	2.7%
9	Payments Interactive	41,128	1.7%
10	Tiki Taka	37,072	1.5%
11	Stanislav Kabin	36,893	1.5%
12	Safe Software	36,602	1.5%
13	Vetaform Developments	36,001	1.5%
14	Outbrowse	35,832	1.4%
15	appbundler.com	34,895	1.4%
16	Rodion Veresev	34,696	1.4%
17	Mari Mara	31,031	1.3%
18	Firseria	29,940	1.2%
19	Give Away software	26,541	1.1%
20	Jelbrus	23,457	0.9%

Table 11: Top 20 publishers in the feed of 11M samples by number of samples and percentage over all samples signed and flagged by at least 4 AV engines.

#	PPI Service	Reseller
1	AdGazelle [2]	
2	EarnPerInstall [18]	
3	GuppyGo [23]	
4	Installaxy [24]	✓
5	InstallMonetizer [29]	
6	MediaKings [37]	
7	NetCashRevenue [43]	✓
8	PayPerInstall [48]	
9	PerInstallBox [49]	
10	PerInstallBucks [50]	✓
11	PerInstallCash [51]	
12	PureBits [55]	✓

Table 12: PPI services found through manual analysis on PPI forums and other Internet resources that are not present in our dataset. The reseller data comes from [65].