REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY): 18-05-2012

2. REPORT TYPE: Conference Proceeding

3. DATES COVERED (From - To): -

4. TITLE AND SUBTITLE: An Assessment of Overt Malicious Activity Manifest in Residential Networks

5a. CONTRACT NUMBER: W911NF-09-1-0553

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER: 611103

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

6. AUTHORS
Gregor Maier, Anja Feldmann, Vern Paxson, Robin Sommer, Matthias Vallentin

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
University of California - Santa Barbara
Office of Research
The Regents of the University of California
Santa Barbara, CA 93106 -2050

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S): ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S): 56142-CS-MUR.7

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT
While conventional wisdom holds that residential users experience a high degree of compromise and infection, this presumption has seen little validation in the way of an in-depth study. In this paper we present a first step towards an assessment based on monitoring network activity (anonymized for user privacy) of 20,000 residential DSL customers in a European urban area, roughly 1,000 users of a community network in rural India, and several thousand dormitory users at a large US university. Our study focuses on security issues that overtly manifest in such

15. SUBJECT TERMS
cybersecurity, hacking, network, security

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UU	UU	UU	UU		Richard Kemmerer
					19b. TELEPHONE NUMBER: 805-893-4232

Standard Form 298 (Rev 8/98)
Prescribed by ANSI Std. Z39.18

Report Title

An Assessment of Overt Malicious Activity Manifest in Residential Networks

ABSTRACT

While conventional wisdom holds that residential users experience a high degree of compromise and infection, this presumption has seen little validation in the way of an in-depth study. In this paper we present a first step towards an assessment based on monitoring network activity (anonymized for user privacy) of 20,000 residential DSL customers in a European urban area, roughly 1,000 users of a community network in rural India, and several thousand dormitory users at a large US university. Our study focuses on security issues that overtly manifest in such data sets, such as scanning, spamming, payload signatures, and contact to botnet rendezvous points. We analyze the relationship between overt manifestations of such activity versus the "security hygiene" of the user populations (anti-virus and OS software updates) and potential risky behavior (accessing blacklisted URLs). We find that hygiene has little correlation with observed behavior, but risky behavior—which is quite prevalent—more than doubles the likelihood that a system will manifest security issues.

Conference Name: Proc. Eighth Conference on Detection of Intrusions and Malware & Vulnerability Assessment

Conference Date: July 07, 2011

An Assessment of Overt Malicious Activity Manifest in Residential Networks

Gregor Maier[1,2], Anja Feldmann[2], Vern Paxson[1,3],
Robin Sommer[1,4], and Matthias Vallentin[3]

[1] International Computer Science Institute, Berkeley, CA, USA
[2] TU Berlin / Deutsche Telekom Laboratories, Berlin, Germany
[3] University of California at Berkeley, CA, USA
[4] Lawrence Berkeley National Laboratory, Berkeley, CA, USA

Abstract. While conventional wisdom holds that residential users experience a high degree of compromise and infection, this presumption has seen little validation in the way of an in-depth study. In this paper we present a first step towards an assessment based on monitoring network activity (anonymized for user privacy) of 20,000 residential DSL customers in a European urban area, roughly 1,000 users of a community network in rural India, and several thousand dormitory users at a large US university. Our study focuses on security issues that *overtly manifest* in such data sets, such as scanning, spamming, payload signatures, and contact to botnet rendezvous points. We analyze the relationship between overt manifestations of such activity versus the "security hygiene" of the user populations (anti-virus and OS software updates) and potential risky behavior (accessing blacklisted URLs). We find that hygiene has little correlation with observed behavior, but risky behavior—which is quite prevalent—more than doubles the likelihood that a system will manifest security issues.

1 Introduction

Conventional wisdom holds that residential users experience a high degree of compromise and infection, due to a dearth of effective system administration and a plethora of users untrained or unwilling to observe sound security practices. To date, however, this presumption has seen little in the way of systematic study. Such analysis can prove difficult to conduct due to major challenges both in obtaining access to monitoring residential activity, and in determining in a comprehensive fashion whether a given system has been subverted given only observations of externally visible behavior.

In this paper we develop first steps towards such an investigation. We examine the network activity of 20,000 residential DSL customers within a European urban area, roughly 1,000 users of a community network in rural India, and several thousand dormitory users at a large US university. As a baseline, we include in our assessment the same analyses for Internet traffic seen at a large research laboratory that has a track record of effectively protecting its roughly 12,000 systems without employing a default-deny firewall policy.

We in addition investigate the relationship between problems flagged by our analyses and the level of apparent "security hygiene" in our populations. One may expect that users who perform regular OS updates and deploy anti-virus software are less likely to have their systems compromised. Likewise, one might presume that users who partake in risky behavior have more security problems. To check these assumptions, we examine the end-user systems in our data sets for signs of regular operating system updates, anti-virus deployments, and contacts to URLs blacklisted by Google's Safe Browsing API [8].

We emphasize that given our data we can only assess malicious activity that *overtly manifests* in network traffic. Clearly, this reflects only a (perhaps quite small) subset of actual security problems among the users. Thus, what we provide constitutes only a first step towards understanding the nature and prevalence of such problems. Given that caution, we frame some of our more interesting findings as:

- In all of our environments, only a small fraction of hosts manifest malicious activity.
- We do not find significant differences between the environment in rural India and the other residential environments, though the former poses a number of analysis difficulties, so this finding is only suggestive.
- While OS software updates and anti-virus technology are widely deployed, we do not find that their use correlates with a lower degree of overt malicious activity.
- We observe frequent risky behavior, and users exhibiting such risky behavior are roughly twice as likely to manifest malicious activity.

We structure the remainder of this paper as follows. After introducing our data sets and methodology in §2, we study security hygiene in §3 and then investigate malicious activity in §4. We present related work in §5. Finally, we discuss our results and present avenues for future work in §6.

2 Methodology

We start our discussion by summarizing the specifics of the data sets we captured in the four network environments we examine in this study. We then discuss how, using these traces, we identify the operating systems used by individual hosts. Finally, we present the metrics we deploy for detecting overt malicious activity.

2.1 Data Sets

We started our work by recording network traces in all environments. However, due to high traffic volume it did not prove feasible to record the full network data stream for an extended period of time in most environments. We therefore leveraged Maier et al.'s "Time Machine" system [13] to reduce the stored amount of data. The Time Machine records only the first N bytes of each connection, discarding the remainder. For example, in the ISP environment, using N=50 KB allowed us to record $\approx$ 90 % of all connections in their entirety while only retaining 10% of total volume.[1]

[1] We note that due to operational constraints we used different cutoff values in each environment, as detailed in the corresponding sections.

Table 1. Data sets for European ISP, Univ, and LBNL. We collected all these using the *Time Machine*.

Location	Start	Duration	Total volume	After cutoff applied	Loss
ISP	Mar 13, 2010	14 days	$>$ 90 TB	$>$ 9 TB	0 %
Univ	Sep 10, 2010	7 days	$>$ 35 TB	$>$ 2 TB	0.005 %
LBNL	Apr 29, 2010	4 days	$>$ 25 TB	$>$ 350 GB	0.2 %

Table 2. Data sets for AirJaldi.

Name	Start	Duration	# IPs	Size	Loss
AirJaldi1	Mar 10, 2010	40 h	263	65 GB	0.35 %
AirJaldi2	Mar 13, 2010	34 h	188	31 GB	1.07 %
AirJaldi3	Apr 22, 2010	34 h	261	51 GB	0.44 %

We used the Bro system [14] with Dynamic Protocol Detection [6] to detect and extract application layer headers. Our capturing processes made sure to anonymize the data in accordance with the network operators' policies, and all data classification and header extraction executed directly on the secured measurement infrastructures. Table 1 and Table 2 summarize our data sets.

Lawrence Berkeley National Laboratory. The *Lawrence Berkeley National Laboratory* (LBNL) is a large research institute with more than 12,000 hosts connected to the Internet via a 10 Gbps uplink, and we use it a baseline for our study. Since LBNL offers an open research environment, LBNL's security policy defaults to fairly unrestricted access, with only limited firewalling at the border. LBNL's security team actively monitors its network for malicious activity, deploying a toolbox of strategies including detectors for scanning and spamming. Systems detected as compromised are quickly taken off the network. Thus, we expect to find little malicious activity in this environment.

We analyze one 4-day packet-level trace. The trace covers two weekdays and a weekend, with a total of 7,000 hosts active during that period. Large individual flows are common at this site, so the Time Machine proved especially effective at reducing volume of collected data. We ran it with a cutoff value of 25 KB. During our recording interval, the capture mechanism reported 0.2 % of all packets as dropped.

In terms of application protocol mix, we find that HTTP contributes about 42 % of bytes at LBNL and SSH 21 %. About 23 % of the traffic remains unclassified.

European ISP. Our largest data set represents a 14-day anonymized packet-level trace of more than 20,000 residential DSL lines, collected at an aggregation point within a large European ISP. The ISP does not employ any traffic shaping or blocking (such as filtering outbound SMTP connections), providing us with an unobstructed view of potential malicious activity.

We used the Time Machine with a cutoff value of 50 KB. Since we employed Endace monitoring cards and discarded the bulk of the data, we did not experience any packet loss. In addition, we also have access to several further day-long traces for comparison purposes.

This ISP data set includes meta-data associating the anonymized IP addresses in the trace with the corresponding (likewise anonymized) DSL line IDs, which enables us to distinguish customers even in the presence of the frequent address reassignment employed by this ISP [11]. For our study, we thus use these line IDs as our basic analysis unit.

Furthermore, the NAT detection approach developed in [12] works well in this environment and allows us to reliably identify the presence of NATs on any of the DSL lines. In addition, the NAT detection approach enables us to estimate the number of hosts connected at a DSL line. We find that 90 % of the DSL lines use NAT and that 46 % connect more than one device to the Internet.

The application protocol mix in the monitored ISP network is dominated by HTTP ($>$ 50% of the total volume), while the prevalence of peer-to-peer applications is rather small (15%). NNTP accounts for roughly 5%, and less than 15% of the traffic remains unclassified but is likely P2P.

Indian community network.. The AirJaldi [1] wireless network is a non-profit community network in the Himalayas of India. Using approximately 400 wireless routers, it covers a radius of 80 km in and around the city of Dharamsala. AirJaldi connects thousands of users and machines with two leased lines from broadband ISPs, which provide a total uplink capacity of 10 Mbps. The majority of the rural population accesses the Internet via publicly shared machines in cybercafes or libraries. In addition, some residential users connect to the network with individually administered machines.

In the AirJaldi network, a single IP address is assigned to each "customer site", such as a specific building, library, or village. Customer sites can in turn provide connectivity to anywhere from one to several hundred systems, and the larger ones typically employ a multi-tiered NAT architecture on their inside, with NAT gateways connecting further NAT gateways.

Due to this architecture, we cannot directly distinguish individual systems at our monitoring point which is located at the central uplink router. Likewise, the NAT detection approach we use for the European ISP cannot reliably determine the number of hosts behind the larger gateways. We therefore report only aggregate findings for malicious activity and risky behavior for the AirJaldi environment.

However, to get an idea of the size of the user population, we can still estimate the number of *active* individual users by extracting (anonymous) user ID cookies found in HTTP traffic: the IDs sent to `google.com`, Doubleclick, and Google Analytics consistently indicate that we are seeing at least 400–600 users in each trace.

In our traces, HTTP dominates AirJaldi's application protocols mix, accounting for 56–72 % of the total traffic volume, similar to the European ISP. However, in contrast to the ISP, interactive communication protocols (instant messaging, VoIP, Skype, etc.) account 2.5–10 % at AirJaldi, while NNTP and P2P are not prevalent.

University Dormitories. We also examine network traffic recorded at dormitories of a major US University, where our monitoring point allows us to observe all their external traffic. While the dormitory network generally employs a default-open firewall, it deploys a custom "light-weight" Snort setup configured to detect a small set of recent malware. Users flagged by Snort are automatically moved over to a containment network from where they can then only access resources for disinfecting their systems.

Note that this containment prevents such victims from further triggering our malicious activity metrics once the dorm setup has restricted their access.

The IP address space for the dorm users is large enough to assign public IP addresses to each local system, with no need for a central gateway NAT. In our data set, we observe about 10,000 active IP addresses. Analyzing user IDs (as we discussed with the AirJaldi network), we find roughly 14,000 distinct ones, suggesting that a single public IP is only used by few machines. We also find that user-to-IP mappings remain mostly stable, and thus we can use IP addresses as our main analysis unit here. Similar to the ISP, however, we cannot further distinguish between different hosts/devices connected potentially through a user-deployed NAT gateway.

We note that the NAT detection approach developed in [12] does not work well in this environment. As it leverages IP TTLs, it relies on a well-known hop distance between the monitoring point and the user system, which is not the case in the dorm environment.

We analyze one 7-day packet-trace from the Univ environment, recorded using the Time Machine with a 15 KB cutoff (i.e., smaller than for the European ISP). The application protocol mix in the dorm network is dominated by HTTP, which accounts for over 70 % of all bytes, with 20–25 % of traffic unclassified.

2.2 Operating Systems

Malware often targets particular operating systems, and some OSes are perceived as more secure due to architectural advantages or smaller market shares that render them less attractive targets. To assess these effects, we annotate our traces with the operating system(s) observed in use with each DSL line (for the European ISP) or IP address (for the other environments). We determine operating systems by analyzing HTTP user-agent strings reflecting the most popular browsers (Firefox, Internet Explorer, Safari, and Opera).

Doing so, we can identify one or more operating systems on more than 90 % of all DSL lines of the European ISP. Analyzing the operating system mix, we find that 59 % of the lines use only a single version of Windows, while 23 % use different Windows versions. Mac OS is present on 7.6 % of the lines, and used exclusively on 2.7 %. We find Linux on 3.7 % of the lines. However, we also observe that Linux is generally used in combination with other operating systems, and the fraction of lines with only Linux is too small to assess malicious activity separately.

At LBNL, we can identify at least one operating system for 60 % of all active IPs. We find that 70 % of these IPs use only a single version of Windows. We see Macs in use with 19 % of those IPs, and Linux with 8.6 %.

In the Univ environment, we are able to determine operating systems for 73 % of all active IPs. Mac OS is the dominating operating system here, present on 56 % of them. 34 % of IPs use a single version of Windows exclusively, and Linux is present on just over 4 % of IPs.

At AirJaldi, we observe a large number of operation systems per IP due to its extensive NAT deployment and thus cannot further estimate the distribution.

2.3 Manifestations of Compromised Systems

To identify end systems that overtly manifest malicious activity or signs of likely compromise, we search for three behavioral indicators—*address scanning*, *port scanning*, and *spamming*—and also monitor for network-level signatures aimed at detecting three malware families, Zlob, Conficker, and Zeus.

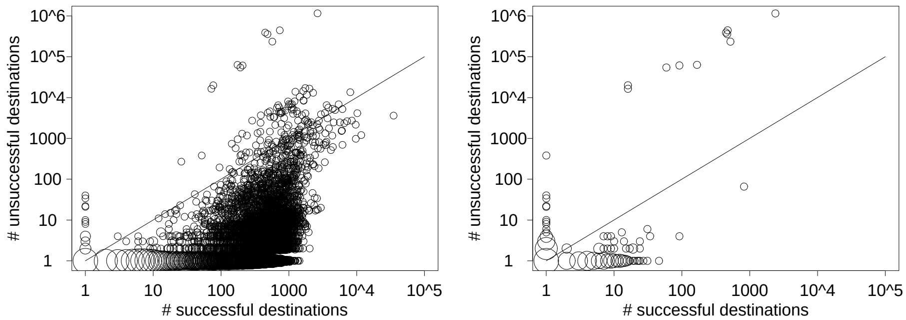
To take advantage of the long duration of our traces, we analyze and report manifestations of malicious activity first separately for each day of our multi-day traces (European ISP, Univ, and LBNL). We then further aggregate the results by accumulating all activity over increasing trace durations into cumulative data sets. For example, consider a local system (DSL line or IP address) that started to be a scanner on day 4. In the daily data set this system is marked as a scanner on day 4. In the cumulative data set, it is marked as scanner on day 4 as well as for *all following days*, regardless of whether the system is again acting as a scanner on any of the subsequent days or not. In particular, the cumulative data for the last day of a trace reflects the aggregate behavior over the full observation period.

Scanning. Extensive *address scanning* often reflects the precursor of an attack that targets services that answer the scan. Most network intrusion detection systems (NIDS) therefore detect and report such scanners but their detectors typically target *external* hosts probing a monitored network. For our study, however, we instead wish to find *outbound* scanners. This can prove harder to do, as the potential probes are embedded within the host's benign activity.

We found that for our environments, a simple threshold-based scan detector often erroneously flags benign user activity such as web browsers. Detectors that count *failed* connection attempts work better, but still can misfire for P2P-style traffic patterns. We therefore use an approach loosely modeled on that of TRW [10]: we compute the *ratio* of successful vs. unsuccessful TCP connections initiated by a local host. We define a connection as unsuccessful if a client sends a SYN but either receives a RST back or no answer at all.

We find that our data exhibits a sharply bi-modal distribution of success ratios: Connections between a specific pair of local and remote hosts tend to either always succeed or always fail. However, when investigating the fraction of remote destinations *per* local host that tend to fail, we do not find such a clear distribution. Presumably, P2P clients can have similarly large numbers of mostly unsuccessful destinations, thus rendering this approach impractical. We can overcome this ambiguity, however, by limiting our scan analysis to 14 ports that frequently appear in inbound scanning, as identified by the security monitoring in our baseline environment, LBNL. These ports include, e.g., the Windows services 135–139, 445, and 1433. For activity targeting these ports, we indeed find that local systems either have a small (< 20) or large (> 100, often $> 1{,}000$) number of unsuccessful remote contacts; see Figure 1. We thus deem activity as overtly manifesting a compromised local system if a host unsuccessfully attempts to contact > 100 destinations.

Another dimension of scanning regards *port scanning*: probing many ports on a single remote host. Since few benign applications need to contact numerous distinct ports on a single system, we use a simple threshold-based approach: we consider a local



(a) including all ports

(b) only including 14 likely scan ports

Fig. 1. Scatter plot showing number of successful vs. unsuccessful connections with remote IPs per DSL line, computed for a single day of the ISP trace. The size of the circles is proportional to the number of lines having the corresponding ratio.

system to manifest malicious activity if it *unsuccessfully* contacts at least two remote hosts on more than 50 ports within a single day.

Spamming. Another manifestation of malicious activity concerns local hosts sending spam. To find an appropriate indicator for classifying a host as a spammer, we examined SMTP activity of local systems in terms of how many remote SMTP servers they contact. The number of distinct SMTP servers can be an indicator since we do not expect many users to run their own SMTP servers, but instead to rely on a (small) number of e-mail providers to deliver their mails. Particularly in the ISP environment, e-mail providers are necessary since the IP range of the monitored DSL lines is dynamically assigned and many SMTP servers generally reject mails from any dynamic addresses unless specifically authenticated. We find that DSL lines consistently contact either less than 25 SMTP servers or more than 100. Indeed, most lines contact less than 10. We ultimately classify a system as a *spammer* if it contacts > 100 distinct SMTP servers in one day.

Known malware families. We can also observe manifestations of host compromise by flagging activity related to known malware families. To do so, we focused on Zlob, Conficker, and Zeus.

The Zlob malware family [24] changes the DNS resolver settings of infected hosts (and even broadband routers), altering them to a well-known set of malicious remote systems. We thus classify a system as reflecting a Zlob infection if it uses one of those known DNS resolvers. We note that Zlob targets both Windows and Mac systems, subverting systems using social engineering in the guise of a fake codec download.

Another malware family that has attracted much attention is Conficker [15]. For April 2010, the Conficker Working Group [4] reported 5–6 M distinct Conficker A+B infected IPs and 100–200 K distinct Conficker C infected IPs. Note however that the number of IPs does not necessarily provide a good estimation of the population size due to aliasing. The group estimates the true population size to be between 25–75 % of the IP addresses seen per day.

To find rendezvous points, Conficker A and B generate a list of 250 different domain names each day (Conficker C uses 50,000 domains) and then try to resolve them. Since the algorithm for generating the domain names is known [22], we can compute the relevant domain names for our observation periods and check which local hosts try to resolve any of these. To account for potential clock skew on the client machines, we also include the domain names for the days before and after our observation period.

Conficker C sometimes generates legitimately registered domains, due to its frequent use of very short names. To rule out false positives, we therefore only flag a system as manifesting a Conficker infection if it issued ≥ 50 lookups for known Conficker domains. We note that Conficker also uses address-scanning to find other vulnerable machines and spread itself. Thus, we expect to find a correlation between address scanning and Conficker lookups.

We use the Zeus Domainblocklist [25] to identify local systems infected with Zeus. Since the list does not only contain seemingly random domain names but also some indicating scare-ware (e. g., `updateinfo22.com`) or typo squatting (e. g., `google-analytiics.cn`), we require ≥ 4 lookups per system and day to classify a system as infected. We found that fewer lookups often reflect embedded ads or other artifacts not due to compromise. Since we initially lacked a continuous feed of the Zeus domain list, we used a one-day snapshot from Mar 23, 2010 for the ISP trace and AirJaldi1/AirJaldi2. We used another snapshot from May 6, 2010 for LBNL and AirJaldi3. For Univ, we were able to continuously update the list during the measurement period.

Alternative Manifestations of Compromised Systems. We also examined further alternative approaches to detect compromised systems, which however did not turn out to generate sufficiently reliable results.

One obvious approach is to leverage a NIDS' output directly. Consequently, we attempted using a standard signature set: the Emerging Threats rules [7] loaded into Snort [18]. We excluded all rules that are labeled as aiming at finding "inappropriate" activity (e. g., gaming traffic or P2P) as such is not in scope for our study. Still, analyzing a single day of ISP traffic, we get more than 1 million hits, flagging more than 90 % of the monitored local systems. Clearly, such results have to include many false positives.

We can reduce the number of hits by restricting the analysis to examine only outbound activity, and whitelisting rules that are likely to flag benign activity. However, Snort still reports too many alarms to be a reliable indicator for us. The main problems with the Emerging Threats rules are their sheer size (more than 3,500 rules in total), and a general lack of documentation (many rules are not documented at all). In particular, the rules do not include any indication on how tight they are (i. e., what their likelihood of false positives is), nor how severe one should consider the activity they target (e. g., is this rule triggered by a C&C channel or by adware).

Another approach we tried as an indicator of likely system compromise is checking whether a local system's IP address appears on an anti-spam blacklist, such as Spamhaus. However, we find that doing so proves difficult in our settings: address assignments are dynamic and can often reflect numerous different hosts over short time intervals. Thus a single blacklist entry can tar the reputation of many distinct local systems. In addition, some anti-spam providers proactively blacklist *all* dynamically assigned IP address ranges.

3 Security Hygiene and Risky Behavior

To analyze whether residential users are aware of potential hazards and take recommended countermeasures, we analyze *(i)* whether local systems use anti-virus scanners; *(ii)* if they regularly update their OS software (e. g., Windows Update); and *(iii)* whether they download Google's Safe Browsing blacklist.

Most anti-virus and OS software updates are done using HTTP, and they use specific user-agents and/or HTTP servers. Searching for those allows us to classify a local system as a software updater and/or anti-virus user. Likewise, the HTTP servers serving Google's blacklist are well-known, and we can thus identify local systems which use the blacklist. Moreover, we also check if local systems downloading the blacklist still request any blacklisted URLs. Such behavior clearly has to be considered risky. For finding such "bad" requests, we determine whether a requested URL was blacklisted at the time of the request by keeping a local change history of the blacklist over the course of our measurement interval.

We note that me might over-estimate the number of actual OS or anti-virus updates. OS updaters often only check whether an update is available and then ask the user to install it, however the user might decline to do so. Likewise, anti-virus products might download (or check for) new signatures but not use them unless the user has a current license for the anti-virus software. Our analysis flags these cases as anti-virus user and OS updater respectively. However, we cross-checked our results by only considering a local system as anti-virus user if the anti-virus software transferred at least 1 MB and 100 KB of HTTP body data.

3.1 European ISP

Figure 2 shows the fraction of active DSL lines vs. lines performing anti-virus or OS updates. On any given day 57–65 % of DSL lines perform OS software updates, and 51–58 % update their anti-virus signatures (or engines). From the cumulative data we see that over the 14 day observation period up to 90 % of lines check for updates to their OS and 81 % use anti-virus software. This highlights that the user-base in principle exhibits elementary "security hygiene" and performs recommended precautions.

When focusing on DSL lines exhibiting exclusively MacOS activity, we find that up to 2 % do anti-virus updates and up to 78 % perform software updates.

But what about risky behavior such as requesting potentially dangerous URLs? Overall, we find that only about 0.03 % of all HTTP requests appear on Google's Safe Browsing blacklist. However, investigating the per-DSL line behavior, we find that on any given day up to 4.4 % of the DSL lines request at least one blacklisted URL (see Figure 3). Across the whole 14-day period we see that a striking 19 % of lines request at least one blacklisted URL.

To check if the browser may have warned the user about doing these requests, we next examine if the user-agent placing them actually downloaded the blacklist from

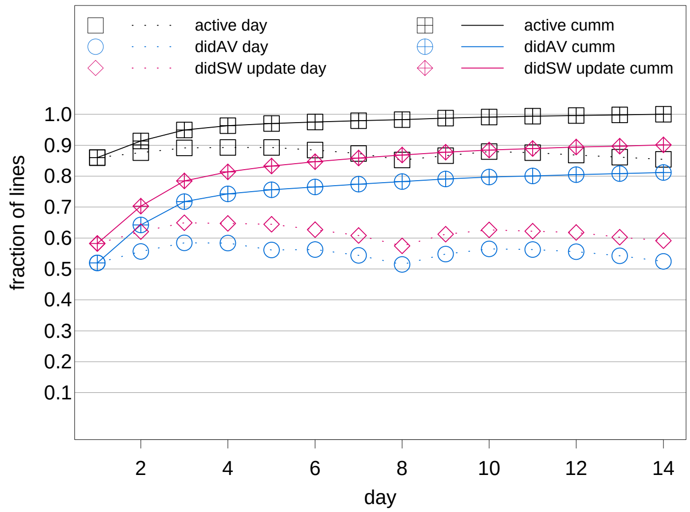


Fig. 2. Fraction of active lines, lines with anti-virus updates, and software updates for daily and cumulative data sets from the ISP.

Google earlier. For doing so, we distinguish three cases when a user accesses a blacklisted URL:

– notDownloaded: The browser did not download the blacklist at all, thus the user could not have been warned.
– beforeBlacklisted: The browser downloaded the blacklist, however the URL was requested *before* it appeared on the blacklist, thus the user could likewise not have been warned.[2]
– whileBlacklisted: The browser downloaded the blacklist and requested the URL while it was blacklisted. That is, the user placing the request (presumably by clicking on a link) *should have been warned* by the browser that the URL is considered malicious.

In Table 3 we list for each combination of cases the fraction of DSL lines exhibiting this combination. We find that for the majority of lines requesting blacklisted URLs the browser either did not download the blacklist (notDownloaded), or the request occurred before the URL was blacklisted and thus the browser could not have warned the user about the URLs potential malignity. However, we find a significant number, 33.6 %, of DSL lines that request blacklisted URLs even though the users should have been warned by their browsers (any combination in which whileBlacklisted is present). When we investigate how often DSL lines request URLs that are on Google's blacklist, we

[2] We allow for a 1 hr grace period, relative to the time when a URL has appeared on our local copy of the list. Google suggests that browser developers implement a 25–30 minute update interval for the blacklist. Furthermore, Google states that a warning can only be displayed to the user if the blacklist is less than 45 minutes old, thus forcing browsers to regularly update the list.

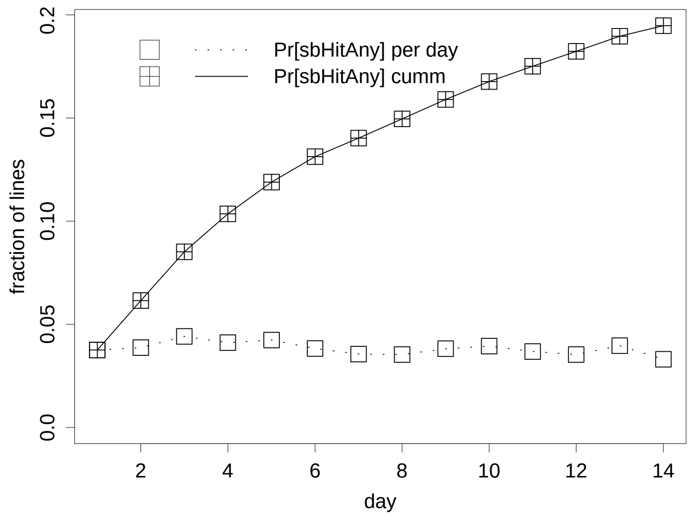


Fig. 3. Fraction of lines requesting URLs blacklisted by Google.

Table 3. Possible cases when DSL lines access blacklisted URLs for ISP dataset. 100 % represents all lines requesting at least one blacklisted URL.

Fraction	Cases
43.8 %	notDownloaded
19.5 %	beforeBlacklisted
16.9 %	whileBlacklisted
11.0 %	whileBlacklisted, beforeBlacklisted
3.0 %	notDownloaded, whileBlacklisted
2.7 %	notDownloaded, whileBlacklisted, beforeBlacklisted
3.1 %	notDownloaded, beforeBlacklisted

find that 53 % of such lines do so on only a single day. This indicates that although many lines request blacklisted URLs, they do not request such URLs every day.

3.2 University

Analyzing security hygiene at the Univ environment, we find that 78 % of all local IP addresses perform OS software updates over the 7-day trace duration, and that 60 % perform anti-virus updates. While these numbers are lower than the numbers for the European ISP, we note that we observe significantly more IP addresses on which only MacOS systems (and no Windows systems) are active than at the European ISP. For IP addresses on which we only observe Macs, we find that 45 % perform anti-virus and 73 % perform software updates.

When we turn to risky behavior we find that after the 7-day observation period roughly 20 % of local IP addresses have requested URLs blacklisted by Google. This is higher than at the European ISP, where we reach 19 % after 14 days (after the first

7 days the European ISP environment reaches 14 %). Similar to the ISP we find that 36.4 % of the IPs requesting blacklisted URLs do so although the browser should have warned the user.

3.3 AirJaldi

Comparing security hygiene at the European ISP with the AirJaldi network in India, we find them quite similar. In terms of HTTP requests, we find that at AirJaldi approximately 0.02 % are blacklisted by Google (European ISP: 0.03 %), and 0.5–1.12 % are related to anti-virus software (European ISP: 1 %). For OS software updates, the numbers differ more: up to 2.8 % of all HTTP requests at AirJaldi, vs. only 0.3 % at the European ISP. Assessing security hygiene on a per host basis is difficult in this environment, however, given the layered NAT structure. We find that 29.6 %–40.7 % of the observed sites at AirJaldi perform anti-virus and OS software updates, and 3.8–5.4 % request blacklisted URLs. Recall, however, that each site can connect anywhere between a handful and several hundred users.

3.4 LBNL

We next turn to security hygiene at LBNL. We find relatively few hosts there updating anti-virus software (24 %) or operating systems (31 %). This can be explained however by the fact that LBNL uses centralized, internal update servers. The operating system distribution also differs notably from the ISP environment, with significantly more Linux and Mac hosts in use at LBNL. Turning to risky behavior, we find that only 0.01 % of all HTTP requests are blacklisted. In terms of the fraction of hosts requesting blacklisted URLs, we also find a smaller number than at the European ISP: up to 0.92 % of hosts per day and less 1.25 % overall. However, we find that users at LBNL still request blacklisted URLs even though they should have been warned by their browser; 23.4 % of IPs requesting blacklisted URLs do so.

4 Malicious activity

After finding that even though users appear to be security aware and take appropriate precautions they still engage in risky behavior, we now use our indicators for identifying malicious behavior. Moreover, we also study the influence of security awareness, risky behavior, NAT deployment, and operating systems on infection probability. We emphasize that the overall estimate of malicious activity that we derive can only be a lower bound of the total present in these environments.

We start by studying the characteristics of the DSL customers of the European ISP since that is our richest data set. We then compare the results from the European ISP with those from the university dorms (Univ), the AirJaldi community network, and the LBNL setting, as applicable.

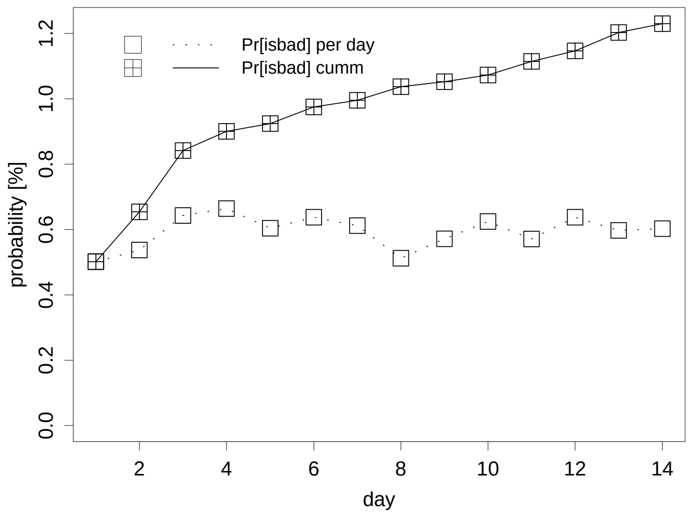


Fig. 4. Probability that a line triggers malicious activity indicators (isbad) for the ISP, shown separately for the daily and cumulative data sets.

4.1 European ISP

Figure 4 shows the likelihood that a DSL line triggers any of the malicious activity indicators for the European ISP. We find that both on a per-day basis as well as overall only a small fraction of local systems manifest overtly malicious activity; $< 0.7\,\%$ and $< 1.3\,\%$, respectively. Moreover, these percentages do not vary much across days.

However, even though the cumulative probabilities remain small, they are increasing over our 14-day period. This indicates that over time we are able to identify more lines that manifest malicious activity, and it may imply that longer observation periods would reveal even more lines manifesting such activity.

We find that *overall* the malware families and spammers contribute most to the observed manifestations of compromised systems, while scanners are less prominent; see Table 4. On a *per-day* basis, only the malware-based indicators are prominent.

More than 44% of the spammers are only active on a single day, i. e., they trigger the indicator only on a single day. In contrast, only 11% of the scanning activity is limited to a single day. On average (mean and median) scanners are seen for 4 days.

For most indicators, we observe a difference between the mean number of days that the indicator is triggered and the median number of days. This indicates that there is no consistent behavior by the local systems manifesting malicious activity. Indeed, an in-depth analysis reveals that some spammers and scanners start their malicious behavior as soon as the DSL line becomes active, while others stop temporarily or are only active for a short period. The fact that the malware families typically trigger on 5–8 days (mean) confirms that the bots engage in activity on a regular basis. However, malicious activity such as port scanning or spamming seems to be limited to sub-periods.

We find only a small overlap among the lines that trigger any of the indicators. Most lines that manifest malicious activity trigger only a single indicator (92 %); about 7 %

Table 4. Probability that a DSL line triggers a malicious activity indicator. The daily numbers summarize the range of the probability values per day. To estimate the persistence of the activity, we include the mean/median number of days that each indicator triggered and the percentage of lines for which the indicator triggered only on a single day.

	Probability		Activity prevalence	
Indicator	daily prob.	cumm. prob.	mean / median	only single day activity
Spam	0.03–0.10 %	0.25 %	3.6 / 2 days	44 %
Scan	0.01–0.06 %	0.09 %	4.3 / 4 days	11 %
Port Scan	0.01–0.03 %	0.06 %	3.5 / 2 days	39 %
Zlob	0.13–0.19 %	0.24 %	8.4 / 10 days	10 %
Conficker	0.17–0.26 %	0.23 %	6.5 / 6 days	27 %
Zeus	0.07–0.15 %	0.28 %	4.9 / 2 days	38 %
Total	0.50–0.66 %	1.23 %	5.9 / 4 days	28 %

trigger two. There is some correlation (0.227) between lines that manifest Conficker and scanning activity, as we would expect due to Conficker's regular network scans in search of further vulnerable hosts. We also observe a correlation of 0.109 between DSL lines triggering the Spam and Zeus metrics.

Next we check whether the likelihood that a local system manifests malicious activity is influenced by other parameters.

Security Hygiene. We start our investigation by examining the impact of anti-virus deployment and OS software updates. Surprisingly, we do not see any strong effects in this regard. Anti-virus software does not noticeably reduce the likelihood of a DSL line manifesting malicious activity (1.10 % with anti-virus vs. 1.23 % without, considering all indicators). We cross-check these numbers for a potential over-estimation by only considering DSL lines that transferred at least 100 KB (1 MB) of anti-virus HTTP payload, we find that still 1.06 % (0.82 % for 1 MB) of lines trigger our metrics. While the latter is a drop of roughly 30 %, anti-virus software still seems less effective than one might presume. Likewise, we do not observe any particular impact of performing OS updates. We verified that these findings are not biased by NATed DSL lines with multiple hosts. That is, even for lines that do not show multiple hosts, the likelihood of infections is not significantly changed by using anti-virus or doing updates.

Google Blacklist. Given the large fraction of DSL lines requesting blacklisted URLs, we next study whether such behavior increases the chance of manifesting malicious activity (see Figure 5). Indeed, we find that the probability of a line triggering any of our indicators rises by a factor of 2.5, up to 3.19 % (from 1.23 %) if we also observe the same line requesting at least one URL found on Google's blacklist. While in absolute terms the elevated level still remains low, our indicators only track *overt* manifestations of security issues, so the more salient takeaway concerns the more-than-doubled rate of host compromise for systems with poor "security hygiene".

NAT. Next, we evaluate whether NAT influences the probability of triggering our malicious activity indicators. We find that DSL lines that connect multiple hosts are 1.5 times more likely to trigger one (1.81 %). Lines on which no NAT gateway is present are as likely to trigger an indicator as all DSL lines.

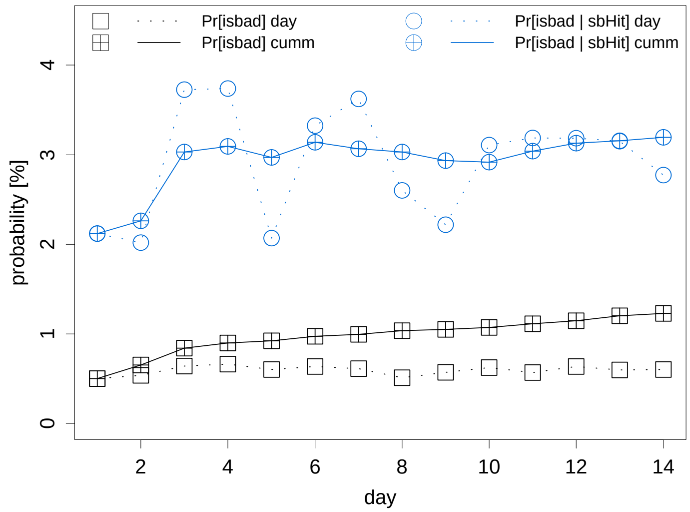


Fig. 5. Probability that a line triggers any malicious activity indicator (isbad) given it requested a blacklisted URL (BlkLst) for ISP.

Malicious activity on Macs. Next, we turn to DSL lines on which all traffic is from Macs (2.7 % of all lines) and evaluate their likelihood of manifesting malicious activity. When analyzing the full 14-day observation period, we find that Mac lines trigger one of our indicators at less than half the rate of all systems (0.54 % vs. 1.23 %), and they do so via a single indicator, infection with Zlob. In this regard, however, Macs are more than *twice* as likely to manifest an infection as all systems (0.24 %).

These findings are somewhat difficult to interpret. They appear to suggest that in general, Mac systems fare better than Windows systems in terms of resisting infection—but possibly only because most infection targets only the latter, and when infection also targets Macs, they may in fact be more vulnerable, perhaps due to less security vigilance.

4.2 University

We next turn to manifestations of malicious activity in the Univ dorm environment. As the dorm network provides central DNS resolvers to all local systems, we are not able to observe DNS requests from local systems unless they specifically configure a different, external DNS resolver. This implies that we cannot use our DNS-based indicators (Conficker and Zeus) for directly identifying infected local systems. We are however able to observe DNS requests from the central resolvers, allowing us to recognize whether any local systems are infected. Analyzing the DNS requests in Univ's traffic, we find no Conficker lookups at all, indicating that no hosts infected with Conficker were active (outside the containment network) during our measurement interval. We do however observe lookups for Zeus domains.

When using the other indicators, we observe that only a few local systems (0.23 %) manifest overtly malicious activity during our 7-day observation period. At the European ISP, 1 % of DSL lines manifest such activity over 7 days (1.23 % over 14 days), and 0.8 % excluding Zeus. In contrast to the European ISP, we find that port scanners at Univ contribute most to the overall number of local systems manifesting malicious activity. Scanners trigger on 0.13 % of local systems and all other indicators combined trigger on 0.1 % of local systems.

We do not observe any local systems that trigger more than one of our malicious activity indicators. This is however not surprising given that we cannot observe Conficker and Zeus activity. Recall that at the European ISP, these two tend to correlate most with other indicators.

Similar to the European ISP, we find that risky behavior (requesting blacklisted URLs) nearly doubles the rate of manifesting malicious activity, with the 0.23 % overall figure rising to 0.4 % in that case. Again, we find that neither OS software nor anti-virus update activities affect the rate at which security problems manifest.

4.3 AirJaldi

	AirJaldi 1	AirJaldi 2	AirJaldi 3
Site 1	**ZeuS** AV SW	**ZeuS** AV SW	**ZeuS** AV SW
Site 2	**Conficker(799)** SW	**Conficker(275)** SW	**Spam** BLK AV SW
Site 3	BLK AV SW	AV SW	**Scan** BLK AV SW
Site 4	X	X	**Spam** AV SW
Site 5	X	X	**Spam** BLK SW
Site 6	X	X	**Spam** BLK AV
Site 7	X	X	**Spam** BLK AV
Site 8	BLK AV SW	**Scan** BLK AV SW	BLK AV SW
Site 9	BLK SW	**Spam** BLK AV SW	BLK AV SW
Site 10	**Conficker(291)** AV SW	AV SW	AV SW
Site 11	**Spam? Scan**	X	X
Site 12	**Scan** BLK AV SW	AV	AV

AV = anti-virus SW = software update BLK = Blacklist hit
Shaded background = malicious activity

Fig. 6. Summary of malicious activity and security hygiene for all AirJaldi traces.

Within the Indian AirJaldi network, we observe very limited malicious activity. However, given that AirJaldi uses a multi-tiered NAT hierarchy, we can only explore malicious activity by customer site rather than per end-system. Within each trace we observe between 188 to 263 active customers sites. Each customer site can connect multiple, possibly hundreds, of hosts. However, exploring the number of destinations for scanners and spammers for each flagged site, we find that they are well within the range of what we observe at the European ISP. Therefore, we conclude that each of the reported scanning and spamming sites is likely due to a small number of infected hosts. Indeed, most likely it is due to a single one.

Across all three traces only a total of 12 customer sites triggered any of our indicators. Figure 6 summarize the results, listing all 12 customer sites that triggered any of our indicators. An "X" indicates that a site was not observed in a particular trace. We also further annotate each site with labels based on whether any of its systems requested blacklisted URLs (*BLK*), or updated anti-virus (*AV*) or OS software (*SW*).

A detailed look at the spammers shows that *site 11* contacted 56 remote SMTP servers, less than our cutoff of 100. However, since no other site contacts more than 10 SMTP servers, we flag this site as a likely spammer (Spam?) rather than a certain spammer (Spam).

Only two sites trigger our malicious activity indicators in more than one trace. We never observe more than 7 sites manifesting malicious activity in any single trace, and in only one case do we find multiple forms of malicious activity at a single customer site. We note that *sites 4–7* are in the same /24 network and could potentially be a single customer site using multiple IP addresses over the trace duration.

Even though we cannot reliably determine the number of infected hosts at each customer site, we can attempt to estimate it. For Conficker this is possible since each infected host will in general generate 250 DNS lookups per day. Thus we estimate the number of infected hosts by dividing the total number of Conficker lookups by 250. We list the number of Conficker lookups per site in parentheses (Conficker(n)) in Figure 6.

Given the inability to identify end hosts, we cannot soundly correlate activity levels, anti-virus, OS software updates, or Google blacklist hits with malicious activity.

4.4 LBNL

At LBNL, our indicators trigger only for a small number of benign hosts. Not surprisingly, local mail servers are reported as "spammers"; and hosts deployed by the Lab's security group for penetration testing purposes are flagged as "scanners". While confirming that our detectors are working as expected, such false alarms are unlikely to occur in the residential environments. Other than these, we do not find any further malicious activity reported at LBNL.

5 Related Work

In our study, we examine several manifestations for malicious activity including scanning and spamming which are established indicators of compromised systems. Most network intrusion detection systems (NIDS), such as the open-source systems Snort [18] and Bro [14], use simple threshold schemes to find scanners. Bro also provides a more sensitive detector, *Threshold Random Walk* [10], that identifies scanners by tracking a system's series of successful and failed connection attempts. However, existing detectors do not work well for finding *outbound* scans, as needed for our study.

Spamming is often countered by blocking known offenders via DNS-based blacklists, such as SORBS [19] or Spamhaus [21]. However, due to the high IP address churn we experience (for example, due to DSL lines frequently changing their IP address [11]), such blacklists do not provide us with a reliable metric. Furthermore, many blacklists include the *full* dynamic IP space. Ramachandran et al. [17] identify spammers by observing the destinations they try to connect to. This eliminates the need for content inspection. The Snare system [9] extends this approach by building a reputation-based engine relying on additional non-payload features. These approaches, however, deploy clustering algorithms, and thus rely on suitable training sets, which are not available for our data set.

We also check our data sets for indicators of specific malware, all of which are analyzed and tracked in detail by other efforts: *Conficker* [4, 15]; Zlob [24]; and Zeus [25]. For example, Bailey et al. [2] survey botnet technology and Dagon et al. [5] examine malware that changes a client's DNS resolver, including the Zlob trojan.

Carlinet et al. [3] ran Snort on DSL traffic from about 5,000 customers of a French ISP to study what contributes to a user's risk of being infected with malware. For their study, Carlinet et al. simply removed the 20 Snort signatures triggering the most alarms. However, they do not further analyze how the remaining signatures contribute to the overall results, what their false-positive rate is, or whether there is relevant overlap between them.

We also leverage Google's Safe Browsing blacklist [8]; the approach used for collecting the blacklist is originally described by Provos et al. in [16].

Stone-Gross et al. [20] try to identify malicious (or negligent) Autonomous Systems (AS). They analyze data from their honeyfarm to identify IP addresses of botnet C&C servers and use four data feeds to find IP addresses of drive-by-download hosting servers and phishing servers. They correlate the information from these sources and the longevity of these malicious servers to compute a "malicious score" for the ASes hosting such malicious servers. ASes with high scores can then be blacklisted or de-peered by other ISPs or network operators. Thus their focus is on the hosting infrastructure rather than individual infected machines.

There has been some work on the prevalence of individual malware. The Conficker Working Group states that 3 million infected hosts is a "conservative minimum estimate", and it cites the study of an anti-virus vendor that finds that 6% of the monitored systems are infected. Weaver [23] estimates the hourly Conficker C population size in March/April of 2009 to average around 400-700 K infections.

6 Conclusion

In this work we have aimed to develop first steps towards understanding the extent of security problems experienced by residential users. Such studies face major difficulties in obtaining in-depth monitoring access for these users. To this end, we have made partial progress by acquiring network-level observations for tens of thousands of users. However, we lack direct end-system monitoring, and as such in this study we must limit our analysis to security issues that *overtly manifest* in network traffic. On the other hand, we were able to obtain such observations from a number of sites that differ in size, geographic location, and nature of the residential users. This diversity then gives us some initial sense of what facets of the residential security experience appear to hold regardless of the specific environment, and what relationships we find behind different facets of each environment and the differing behavior of individual users.

In this regard, our analysis develops a number of findings:

– A typical residential system is unlikely to engage in scanning or spamming, nor to contact known botnet command-and-control servers.
– Residential users generally exhibit good "security hygiene": many of them update their systems regularly and deploy anti-virus software.
– However, such hygiene does not appear to have much impact on the likelihood of becoming infected with malware.
– A significant subset of users exhibit risky behavior: they contact malicious sites even though, as best as we can deduce, their browsers have warned them in advance.
– Such risky behavior roughly doubles the likelihood of becoming infected with malware.

Our range of data sets allows us to also infer *relative* comparisons between different residential environments. Our main finding in this regard is that seemingly quite different sites—a European ISP, a US university dorm complex, and a rural Indian network—all exhibit similar levels of both security hygiene and risky behavior. Fairly assessing levels of overt malicious activity across these environments is challenging, in part due to ambiguity and limited information for the rural Indian network and limited observability (no DNS based metrics) at the Univ dorm environment. However, for both the European ISP and the Univ environment often only a single indicator manifests per local system. In contrast, for our baseline system at LBNL, we do not observe any overt malicious activity, and significantly less risky behavior.

Finally, our work establishes that much more detailed data will be required to build up a fully comprehensive picture of security issues in residential environments. We have shown that *overt* malicious activity is in fact fairly tame for these networks. The next, very challenging, step is to determine how to construct a sound understanding of *covert* malicious activity.

7 Acknowledgments

We would like to thank Yahel Ben-David of AirJaldi for helping us with the AirJaldi network and the anonymous reviewers for their valuable comments. We would also like to thank our data providers: the European ISP, the US University, the AirJaldi network, and the Lawrence Berkeley National Laboratory.

This work was supported in part by NSF Awards CNS-0905631 and NSF-0433702; the U.S. Army Research Laboratory and the U.S. Army Research Office under MURI grant No. W911NF-09-1-0553; a grant from Deutsche Telekom Laboratories Berlin; and a fellowship within the postdoctoral program of the German Academic Exchange Service (DAAD). Opinions, findings, and conclusions or recommendations are those of the authors and do not necessarily reflect the views of the National Science Foundation, the U.S. Army Research Laboratory, the U.S. Army Research Office, or DAAD.

References

1. AirJaldi Network. `http://www.airjaldi.org`.
2. Bailey, M., Cooke, E., Jahanian, F., Xu, Y., and Karir, M. A survey of botnet technology and defenses. In *Proc. Cybersecurity Applications & Technology Conference for Homeland Security* (2009).
3. Carlinet, Y., Me, L., Debar, H., and Gourhant, Y. Analysis of computer infection risk factors based on customer network usage. In *Proc. SECUWARE Conference* (2008).
4. Conficker Working Group. http://www.confickerworkinggroup.org.
5. Dagon, D., Provos, N., Lee, C. P., and Lee, W. Corrupted DNS resolution paths: The rise of a malicious resolution authority. In *Proc. Network and Distributed System Security Symposium (NDSS)* (2009).
6. Dreger, H., Feldmann, A., Mai, M., Paxson, V., and Sommer, R. Dynamic application-layer protocol analysis for network intrusion detection. In *Proc. USENIX Security Symposium* (2006).
7. Emerging Threats. `http://www.emergingthreats.net/`.
8. Google. Google safe browsing API. `http://code.google.com/apis/safebrowsing/`.
9. Hao, S., Feamster, N., Gray, A., Syed, N., and Krasser, S. Detecting spammers with SNARE: spatio-temporal network-level automated reputation engine. In *Proc. USENIX Security Symposium* (2009).
10. Jung, J., Paxson, V., Berger, A., and Balakrishnan, H. Fast portscan detection using sequential hypothesis testing. In *Proc. IEEE Symp. on Security and Privacy* (2004).
11. Maier, G., Feldmann, A., Paxson, V., and Allman, M. On dominant characteristics of residential broadband internet traffic. In *Proc. Internet Measurement Conference (IMC)* (2009).
12. Maier, G., Schneider, F., and Feldmann, A. NAT usage in residential broadband networks. In *Proc. Conf. on Passive and Active Measurement (PAM)* (2011).
13. Maier, G., Sommer, R., Dreger, H., Feldmann, A., Paxson, V., and Schneider, F. Enriching network security analysis with time travel. In *Proc. ACM SIGCOMM Conference* (2008).
14. Paxson, V. Bro: A system for detecting network intruders in real-time. *Computer Networks Journal 31*, 23–24 (1999). Bro homepage: `http://www.bro-ids.org`.
15. Porras, P., Saidi, H., and Yegneswaran, V. An analysis of Conficker's logic and rendezvous points. Tech. rep., SRI International, 2009.
16. Provos, N., Mavrommatis, P., Rajab, M. A., and Monrose, F. All your iFRAMEs point to us. In *Proc. USENIX Security Symposium* (2008).
17. Ramachandran, A., Feamster, N., and Vempala, S. Filtering spam with behavioral blacklisting. In *Proc. ACM Conf. on Computer and Communications Security (CCS)* (2007).
18. Roesch, M. Snort: Lightweight intrusion detection for networks. In *Proc. Systems Administration Conference (LISA)* (1999).
19. SORBS. `http://www.au.sorbs.net`.
20. Stone-Gross, B., Kruegel, C., Almeroth, K., Moser, A., and Kirda, E. FIRE: FInding Rogue nEtworks. In *Proc. Computer Security Applications Conference (ACSAC)* (2009).
21. The Spamhaus Project. `http://www.spamhaus.org`.
22. Universität Bonn. `http://net.cs.uni-bonn.de/wg/cs/applications/containing-conficker/`.
23. Weaver, R. A probabilistic population study of the Conficker-C botnet. In *Proc. Conf. on Passive and Active Measurement (PAM)* (2010).
24. Wikipedia. Zlob trojan. `http://en.wikipedia.org/wiki/Zlob_trojan`.
25. ZeuS Tracker. `https://zeustracker.abuse.ch`.