Towards Measuring and Mitigating Social Engineering Software Download Attacks

Terry Nelms, *Georgia Institute of Technology and Damballa;* **Roberto Perdisci**, *University of Georgia and Georgia Institute of Technology;* **Manos Antonakakis**, *Georgia Institute of Technology;* **Mustaque Ahamad**, *Georgia Institute of Technology and New York University Abu Dhabi*

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

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

Open access to the Proceedings of the 25th USENIX Security Symposium is sponsored by USENIX

一、作者及所属单位

1. Terry Nelms

姓名：Terry Nelms

单位：Georgia Institute of Technology

Damballa, Inc

电子邮箱：tnelms@gatech.edu

个人主页：https://www.researchgate.net/profile/Terry_Nelms

研究方向：　基于多核处理器的网络数据高速处理

网络数据包的内容审查

浏览器交互过程的重构

恶意软件下载的检测

该作者近期发表的相关论文：

- Nelms T, Perdisci R, Antonakakis M, et al. WebWitness: investigating, categorizing, and mitigating malware download paths[C]//24th USENIX Security Symposium (USENIX Security 15). 2015: 1025-1040.
- Neasbitt C, Perdisci R, Li K, et al. Clickminer: Towards forensic reconstruction of user-browser interactions from network traces[C] //Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2014: 1244-1255.
- Battestilli L, Nelms T, Hunter S W, et al. High-performing scale-out solution for deep packet processing via adaptive load-balancing[C] //Local & Metropolitan Area Networks (LANMAN), 2011 18th IEEE Workshop on. IEEE, 2011: 1-6.
- Nelms T, Ahamad M. Packet scheduling for deep packet inspection on multi-core architectures[C]//Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems. ACM, 2010: 21.

单位信息：

- Georgia Institute of Technology

在 2017 泰晤士高等教育世界大学排名中，Georgia Institute of Technology 排名世界第 33，清华大学排名第 35，该校实力由此可见一斑。在网络安全领域，该校在美国排名第 9。

- Damballa

Damballa is an American computer security company focused on advanced cyber threats such as zero-day attacks and advanced persistent threats (APT). The company's system uses massive data sets and machine learning to identify malicious activity based on network behavior, content analysis and threat intelligence.

作者相关研究工作：

该作者主要关注基于多核处理器的网络数据包的高速处理。此外，该作者还关注于网络数据包的内容审查，重构用户浏览器的交互过程以及而已软件下载等方面的研究。上述的相关研究工作是社会工程学软件下载攻击研究的重要组成部分。

2. Robnerto Perdisci

姓名：Robnerto Perdisci

职称：Associate Professor

单位： Department of Computer Science,

University of Georgia and Georgia Institute of Technology

个人主页：http://roberto.perdisci.com/

电子邮箱：perdisci@cs.uga.edu

研究方向：

My research focuses on securing networked systems. I am particularly interested in web security, automating the analysis of security incidents, and defending networks from malware. I often combine systems research with machine learning and large-scale data mining techniques to solve challenging computer and network security problems. I am also interested in broader aspects of networked systems, including Internet-scale measurements, analysis and optimization of systems performance, and the design of networking protocols. In 2012, I received an NSF CAREER award on a project titled "Automatic Learning of Adaptive Network-Centric Malware Detection Models." Here you can find a list of my publications.

作者相关研究工作：

该作者主要关注于基于浏览器的网络攻击，包括：重构用户浏览器的访问踪迹。同时，该作者还关注恶意软件下载方面研究。上述的相关研究工作是社会工程学软件下载攻击研究的重要组成部分。

近期发表的论文：

- [NDSS] Phani Vadrevu, Jienan Liu, Bo Li, Babak Rahbarinia, Kyu Hyung Lee, Roberto Perdisci. "Enabling Reconstruction of Attacks on Users via Efficient Browsing Snapshots". Network and Distributed System Security Symposium, NDSS 2017 (acceptance rate 16.1% = 68/423) [to appear] [USENIX Sec] Terry Nelms, Roberto Perdisci, Manos Antonakakis, Mustaque Ahamad. "Towards Measuring and Mitigating Social Engineering Software Download Attacks." USENIX Security Symposium, 2016. (acceptance rate 15.6% = 72/463)
- [CCS] Christopher Neasbitt, Bo Li, Roberto Perdisci, Long Lu, Kapil Singh, Kang Li. "WebCapsule: Towards a Lightweight Forensic Engine for Web Browsers." ACM Conference on Computer and Communications Security, ACM CCS 2015. (acceptance rate 19.4% = 128/659)
- [USENIX Sec] Terry Nelms, Roberto Perdisci, Manos Antonakakis, Mustaque Ahamad. "WebWitness: Investigating, Categorizing, and Mitigating Malware Download Paths." Usenix Security Symposium, 2015. (acceptance rate 15.7% = 67/426)
- [SIGCOMM] Maria Konte, Roberto Perdisci, Nick Feamster. "ASwatch: An AS Reputation System to Expose Bulletproof Hosting ASes." ACM SIGCOMM 2015 Conference. (acceptance rate 15.6% = 40/256)

- [DSN] Babak Rahbarinia, Roberto Perdisci, Manos Antonakakis. "Segugio: Efficient Behavior-Based Tracking of New Malware-Control Domains in Large ISP Networks." IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2015. (acceptance rate 21.8% = 50/229)
- [WWW] Xinyu Xing, Wei Meng, Udi Weinsberg, Anmol Sheth, Byoungyoung Lee, Roberto Perdisci, Wenke Lee. "Unraveling the Relationship Between Ad-Injecting Browser Extensions and Malvertising." International World Wide Web Conference, WWW 2015. (acceptance rate 14.1% = 131/929)
- [CCS] Christopher Neasbitt, Roberto Perdisci, Kang Li, Terry Nelms. "ClickMiner: Towards Forensic Reconstruction of User-Browser Interactions from Network Traces." ACM Conference on Computer and Communications Security, ACM CCS 2014. (acceptance rate 19.5% = 114/585)

3. Manos Antonakakis

姓名：Manos Antonakakis

职称：Assistant Professor

单位：Georgia Institute of Technology,

School of Computer Science (Adjunct),

School of Electrical and Computer Engineering.

个人主页：http://www.antonakakis.org/

电子邮箱：manos@anTonaKakiS.oRg / manos@gaTeCh.eDu

研究方向：

My main research interests revolve around computer & network security, anomaly detection, data mining and attack attribution. Most of my research projects take place in the Astrolavos Lab, which I proudly run. My students and I founded NetRisk in order to commercialize a portion of our ongoing research projects.

作者相关研究工作：

该作者主要关注 DSN 安全以及恶意软件下载攻击，同时，作者还关注恶意软件下载攻击，上述的相关研究工作是社会工程学软件下载攻击研究的重要组成部分。

近期发表的论文:

- [ESORICS '16] Bharat Srinivasan, Payas Gupta, Manos Antonakakis and Mustaque Ahamad,"Understanding Cross-Channel Abuse with SMS-Spam Support Infrastructure Attribution", In the 21st European Symposium on Research in Computer Security, Heraklion, Greece, September 28-30, 2016.
- [RAID '16] Athanasios Kountouras, Panagiotis Kintis, Chaz Lever, Yizheng Chen, Yacin Nadji, David Dagon, Manos Antonakakis, and Rodney Joffe, "**Enabling Network Security Through Active DNS Datasets**", In the 19th International Symposium on Research in Attacks, Intrusions and Defenses, September 19-21 at Telecom SudParis, Evry, France.
- [USENIX Security '16] Terry Nelms, Roberto Perdisci, Manos Antonakakis, and Mustaque Ahamad, "Towards Measuring and Mitigating **Social Engineering Software Download Attacks**", In the 25th USENIX Security Symposium, 2016.
- [DIMVA '16] Yizheng Chen, Panagiotis Kintis, Manos Antonakakis, Yacin Nadji, David Dagon, Wenke Lee and Michael Farrell, "**Financial Lower Bounds of Online Advertising Abuse**", In the 13th Conference on Detection of Intrusions and Malware & Vulnerability Assessment, Donostia-San Sebastián, Spain, July 7-8, 2016.
- [DIMVA '16] Panagiotis Kintis, Yacin Nadji, David Dagon, Michael Farrell and Manos Antonakakis, "Understanding the Privacy Implications of ECS", In the 13th Conference on Detection of Intrusions and Malware & Vulnerability Assessment, Donostia-San Sebastián, Spain, July 7-8, 2016.
- [Oakland '16] Charles Lever, Robert Walls, Yacin Nadji, David Dagon, Patrick McDaniel, Manos Antonakakis, "Domain-Z: 28 Registrations Later --- **Measuring the Exploitation of Residual Trust in Domains**", In the 37th IEEE Symposium on Security and Privacy, 2016.
- [USENIX Security '15] Terry Nelms, Roberto Perdisci, Manos Antonakakis, Mustaque Ahamad. "WebWitness: Investigating, Categorizing, and **Mitigating Malware Download Paths**." In the USENIX Security Symposium, 2015.
- [DSN '15] Babak Rahbarinia, Roberto Perdisci, Manos Antonakakis. "Segugio: Efficient Behavior-Based Tracking of New **Malware-Control Domains** in Large ISP Networks." In the 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, Rio de Janeiro, Brazil, June 22-25, 2015.
- [DSN '14] Yizheng Chen, Manos Antonakakis, Roberto Perdisci, Yacin Nadji, David Dagon, Wenke Lee. "DNS Noise: **Measuring the Pervasiveness of Disposable Domains in Modern DNS Traffic**." In the 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, June 23 - 26, 2014 Atlanta, Georgia USA.
- [CCS '13] Yacin Nadji, Manos Antonakakis, Roberto Perdisci, Wenke Lee, David Dagon. "Beheading Hydras: Performing Effective **Botnet** Takedowns." In the 20th ACM Conference on Computer and Communications Security, November 4 – 8, Berlin, Germany.
- [LEET '13] Babak Rahbarinia, Roberto Perdisci, Manos Antonakakis, David

Dagon, "SinkMiner: **Mining Botnet Sinkholes for Fun and Profit**", In the 6th USENIX Workshop on Large-Scale Exploits and Emergent Threats, Washington-DC, August 14-16, 2013.

- [ESORICS '13] Phani Vadrevu, Babak Rahbarinia, Roberto Perdisci, Kang Li, Manos Antonakakis. "Measuring and Detecting **Malware Downloads in Live Network Traffic**." In the 18th European Symposium on Research in Computer Security, RHUL, Egham, UK, 2013. (Source code for Amico is here: https://code.google.com/p/amico/)
- [RAID '13] Yacin Nadji, Manos Antonakakis, Roberto Perdisci, and Wenke Lee. "Connected Colors: **Unveiling the Structure of Criminal Networks**." In the 16th International Symposium on Research in Attacks, Intrusions and Defenses, St. Lucia, October 23-25, 2013.
- [NDSS '13] Charles Lever, Manos Antonakakis, Bradley Reaves, Patrick Traynor and Wenke Lee. "The Core of the Matter: Analyzing Malicious Traffic in Cellular Carriers", In the Proceedings of The 20th Annual Network and Distributed System Security Symposium, San Diego, CA, 24-27 February 2013.
- [USENIX Security '12] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee, David Dagon, "**From Throw-Away Traffic to Bots**: Detecting the Rise of DGA-Based Malware", In the 21th USENIX Security Symposium, Bellevue, WA, August 8–10, 2012.
- [ACSAC '11] Yacin Nadji, Manos Antonakakis, Roberto Perdisci, Wenke Lee, "Understanding the Prevalence and Use of Alternative Plans in **Malware with Network Games**", In the Proceedings of The 27th Annual Computer Security Applications Conference, Orlando, FL, December 2011.
- [USENIX Security '11] Manos Antonakakis, Roberto Perdisci, Wenke Lee, Nikolaos Vasiloglou, David Dagon, "Detecting **Malware Domains at the Upper DNS Hierarchy**", in the 20th USENIX Security Symposium, San Francisco, CA, August 8-12, 2011.
- [RAID '10] Manos Antonakakis, David Dagon, Luo Xiapu, Roberto Perdisci, Wenke Lee and Justin Bellmor. "A Centralized Monitoring Infrastructure for Improving **DNS Security**", In the 13th International Symposium on Recent Advances in Intrusion Detection, Ottawa, Ontario, Canada, September 15-17, 2010.
- [USENIX Security '10] Manos Antonakakis, Roberto Perdisci, David Dagon, Wenke Lee and Nick Feamster. "Building a **Dynamic Reputation System for DNS**", In the 19th USENIX Security Symposium, Washington D.C., August 11, 2010. (Recipient of Google Research Award, liaison at Google; Niels Provos.)
- [DSN '09] Roberto Perdisci, Manos Antonakakis, Xiapu Luo and Wenke Lee. "WSEC DNS: Protecting Recursive **DNS Resolvers from Poisoning Attacks**", In the Proceedings of Dependable Computing and Communications Symposium at the International Conference on Dependable Systems and Networks, Estoril, Lispon, June 29 - July 2 2009.
- [NDSS '09] David Dagon, Manos Antonakakis, Kevin Day, Xiapu Luo, Christopher P.Lee, and Wenke Lee. "Recursive **DNS Architectures** and

Vulnerability Implications", In the Proceedings of The 16th Annual Network and Distributed System Security Symposium, San Diego, CA, February 2009.

- [CCS '08] David Dagon, Manos Antonakakis, Paul Vixie, Tatuya Jinmei, Wenke Lee, "Increased **DNS Forgery** Resistance Through 0x20-Bit Encoding", In the 15th ACM Computer and Communications Security Conference, Alexandria, VA, USA, October 2008.

4. Mustaque Ahamad

姓名：Mustaque Ahamad

职称：Professor

单位：College of Computing ,

Georgia Institute of Technology

and New York University Abu Dhabi

个人主页：http://www.cc.gatech.edu/~mustaq/

http://nyuad.nyu.edu/en/academics/faculty/mustaque-ahamad.html

http://www.iisp.gatech.edu/mustaque-ahamad

电子邮箱：mustaq@cc.gatech.edu

研究方向

Ahamad is a Professor of Computer Science at Georgia Tech Institute of Technology, where he held a professorship since 1995. His fields of interest include: Information Security, Distributed Operating Systems, Middleware for Distributed Applications, and Distributed Algorithms. His teaching focus includes courses in operating systems (undergraduate and graduate), distributed systems, information security, secure computer systems and information security laboratory.

作者相关研究工作：

该作者主要关注于电子医疗数据的安全性问题，与社会工程学软件下载攻击研究并无太大关联。

近期发表的论文：

- Phoneypot: Data-driven Understanding of Telephony Threats, Payas Gupta,

Bharat Srinivasan, Vijay Balasubramaniyan, Mustaque Ahamad; (2015)

- Combating Abuse of Health Data in the Age of eHealth Exchange, Musheer Ahmed, Mustaque Ahamad; IEEE International Conference on Healthcare Informatics - ICHI; (2014)
- Augmenting security and accountability within the eHealth Exchange, M Ahmed, M Ahamad, T Jaiswal; IBM Journal of Research and Development; (2014)
- ExecScent: Mining for New C&C Domains in Live Networks with Adaptive Control Protocol Templates, T Nelms, R Perdisci, M Ahamad; USENIX Security; (2013)
- Redactable and auditable data access for bioinformatics research, J Brown, M Ahamad, M Ahmed, DM Blough, T Kurc, A Post, J Saltz; AMIA Summits on Translational Science Proceedings; (2013)
- Enabling robust information accountability in e-healthcare systems, D Mashima, M Ahamad; 3rd USENIX Workshop on Health Security and Privacy; (2012)
- One-time cookies: Preventing session hijacking attacks with stateless authentication tokens, I Dacosta, S Chakradeo, M Ahamad, P Traynor; ACM Transactions on Internet Technology; (2012)
- Protecting health information on mobile devices, M Ahmed, M Ahamad; Proceedings of the second ACM conference on Data and Application Security and Privacy; (2012)
- Enhancing accountability of electronic health record usage via patient-centric monitoring, D Mashima, M Ahamad; Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium; (2012)
- Trust no one else: Detecting MITM attacks against SSL/TLS without third-parties, I Dacosta, M Ahamad, P Traynor; Computer Security – ESORICS; (2012)
- User-centric Identity Management Architecture Using Credential-holding Identity Agents, D Mashima, D Bauer, M Ahamad, D Blough; Digital Identity and Access Management: Technologies and Frameworks, IGI Global; (2011)
- 2011 Index IEEE Transactions on Parallel and Distributed Systems Vol. 22; TS Abdelrahman, T Abdelzaher, D Abramson, M Ahamad, ...; IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS; (2011)
- Improving authentication performance of distributed SIP proxies, I Dacosta, V Balasubramaniyan, M Ahamad, P Traynor; IEEE Transactions ON Parallel and Distributed Systems; (2011)
- CT-T: MedVault-ensuring security and privacy for electronic medical records, DM Blough, L Liu, F Sainfort, M Ahamad; Georgia Institute of Technology; (2011)
- One-time cookies: Preventing session hijacking attacks with disposable credentials; I Dacosta, S Chakradeo, M Ahamad, P Traynor; Georgia Institute of Technology; (2011)

二、研究背景与意义

背景：

- Most modern malware infections happen via the browser, typically triggered by social engineering.
- Because social engineering (SE) attacks target users, rather than systems, current defense solutions are often unable to accurately detect them.
- While there have been numerous studies on measuring and defending against drive-by downloads, little attention has been dedicated to studying social engineering attacks.

意义：

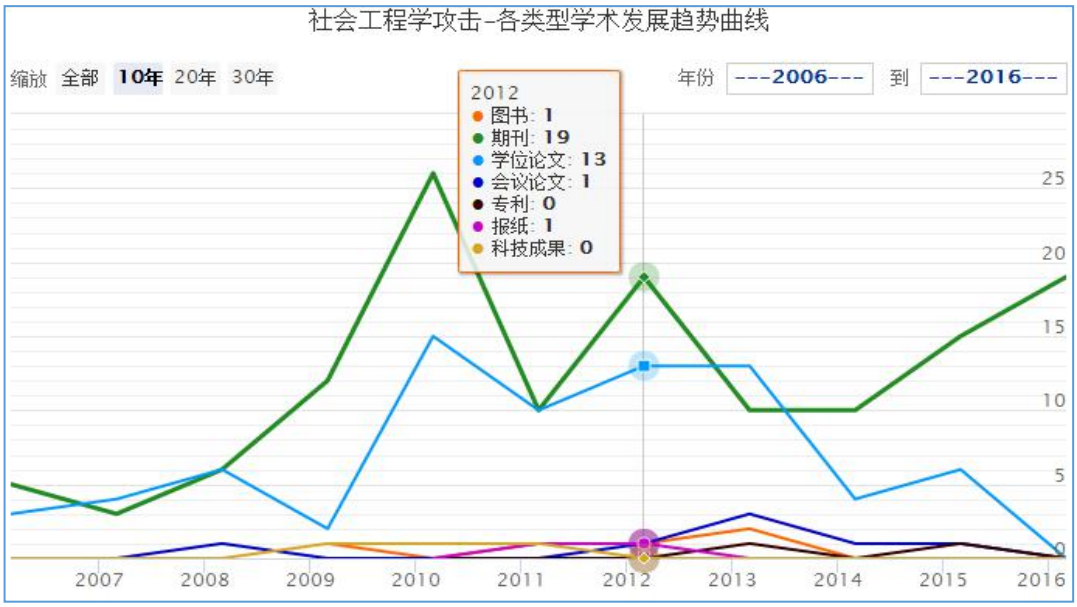
- present the first systematic study of web-based social engineering (SE) attacks.
- shed light on the tactics used in modern attacks.
- focus on the collection, analysis, and categorization of SE download attacks, and on the detection and mitigation of ad-based SE infections.
- create a high quality dataset of labeled SE download attacks.
- gather precious information that may be used to better train users against future SE attacks.

三、研究现状与趋势

研究现状：

目前国内外针对社会工程学网络攻击的研究并不深入，对社会工程学攻击的流程和主要攻击方式缺乏清晰认识，目前防御社会工程学攻击的主要技术手段是对邮件进行检测。在恶意软件下载攻击方面，已经有大量研究，但是不够系统。研究内容大部分关注于一个具体的场景或者一个具体的技术。

趋势：



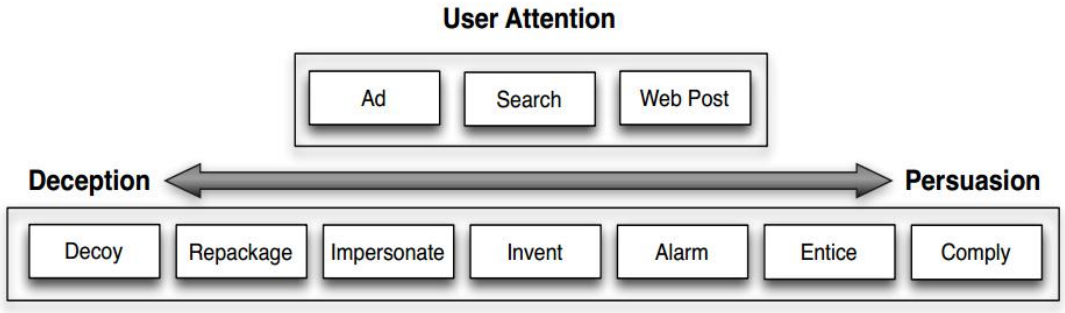
四、论文研究内容

1. Categorizing SE Download Tactics

We analyze the range of deception and persuasion tactics employed by the attackers to victimize users, and propose a categorization system to systematize the in-the-wild SE tactics.

The first step in a SE attack is to get the user's attention. This is accomplished for example by leveraging online advertisement (ad), search engine optimization (SEO) techniques or by posting messages (and clickable links) on social networks, forums, and other sites that publish user-generated content.

After an attacker gains the user's attention, they must convince them to download and install their malicious or unwanted software. This typically involves combining a subset of the deception and persuasion techniques.



- **Decoy**: Attackers will purposely place decoy clickable objects, that will attract users to it.
- **Repackage**: To distribute malicious and unwanted software, attackers

may group benign and PUP (Potentially Unwanted Program) or malware executables together, and present them to the user as a single application.

- **Impersonate**: Using specific images, words and colors can make an executable appear as if it was a known popular benign application.
- **Invent**: Creating a false reality for the user may compel them to download a malicious or unwanted executable.
- **Alarm**: Using fear and trepidation aims to scare the user into downloading (malicious) software that promises to safeguard them.
- **Entice**: Attackers often attempt to attract users to download a malicious or unwanted executable by offering features, content or advantages.
- **Comply**: A user may be (apparently) required to install an (malicious) application before she can continue.

2. Collecting and Labeling SE Attacks

We discuss how we collect software downloads (including malicious ones) from live network traffic and reconstruct their download path. Namely, we trace back the sequence of pages/URLs visited by a user before arriving to a URL that triggers the download of an executable file (we focus on Windows portable executable files). We then analyze the collected software download events, and label those downloads that result

from SE attacks.

Data Collection Approach

We deployed the data collection process described above on a large academic network serving tens of thousands users for a period of two months. Using deep packet inspection, we process the network traffic in real-time, reconstructing TCP connections, analyzing the content of HTTP responses and recording all traffic related to the download of executable files (Windows executables).

When an HTTP transaction that carries the download of an executable file is found, we passively reconstruct and store a copy of the file itself. In addition, we trigger a dump of the traffic buffer, recording all the web traffic generated by the same client that initiated the file download during the past few minutes before the download started. We then process these HTTP transaction captures using the trace-back algorithm walks backwards along this graph to trace back the most likely path.

To avoid unnecessarily storing the download traces related to frequent software updates for popular benign software, we compiled a conservative whitelist consisting of 128 domain names owned by major software vendors.

Overall, during our two month deployment, we collected a total of 35,638 executable downloads.

Automatic Data Filtering

We found that the majority of executable downloads observed on a network are updates, As we are interested in new infections caused by webbased SE attacks, we aim to automatically identify and filter out such software updates.

First, we examine the length of the download path. Software updates tend to come from very short download paths, Conversely, the download path related to SE download attacks usually consists of a number of navigation steps.

For the next step, we review the user-agent string observed in the HTTP requests on the download path. The user-agent string appearing in software update requests is typically not the one used by the client's browser.

Overall, we were able to reduce our download traces dataset by 61%, leaving us with a total of 13,762 that required further analysis and labeling.

Analysis of Software Download Events

We identify a number of statistical features that allow us to discover clusters of download events that are similar to each other and group together similar download events triggered by the same SE attack campaign.

The purpose of this clustering process is simply to reduce the time needed to manually analyze the software download events we collected and minimize the impact of possible noise in the results.

To perform the clustering, we leverage a number of simple statistical features.

- Filename Similarity;
- File Size Similarity;
- URL Structure Similarity;
- Domain Name Similarity;
- Shared Domain Predecessor;
- Shared Hosting;
- HTTP Response Header Similarity;

For each of the 13,762 downloads we compute a feature vector based on the features listed above, and calculate the pairwise distance between these feature vectors. We then apply an agglomerative hierarchical clustering algorithm to the obtained distance matrix. This process produced 1,205 clusters.

Labeling SE Download Attacks

After clustering similar software download events, we manually examine each cluster to distinguish between those that are related SE download attack campaigns, and clusters related to other types of

software download events. This labeling process allows us to focus our attention on studying SE download attacks, and to exclude other types of downloads.

To perform the cluster labeling, each cluster was manually reviewed by randomly sampling 10% of the download events grouped in the cluster, and then performing a detailed manual analysis of the events in this sample set. The actual labeling of SE attacks is performed via an extensive review of each download path.

Overall, among 1,205 clusters in our dataset, we labeled 136 clusters as social engineering. In aggregate, these clusters included a total of of 2,004 SE download attacks.

3. Measuring SE Download Properties

We measure how the SE attack tactics are distributed, according to the categorization system we proposed. According to our dataset, the majority of SE attacks are promoted via online advertisement. Therefore, We also present an analysis of the network-level properties of ad-based SE malware attacks, and contrast them with properties of ad-driven benign software downloads

Popularity of SE Download Attacks

Table 1 shows the number and percentage of SE download attacks for

each tactic employed by the attackers to gain the user's attention. Over 80% of the SE attacks we observed used ads displayed on websites visited by the user. An additional 7% employed both search and ad, whereby the user frst queries a search engine and is then presented with targeted ads based on the search terms.

Table 1: Popularity of SE techniques for gaining attention.

User's Attention	Total	Percentage
Ad	1,616	80.6%
Search+Ad	146	7.3%
Search	127	6.3%
Web Post	115	5.7%

Table 2 shows the popularity of the deception and persuasion techniques we observed in our dataset of SE download attacks. The most popular combination of deception and persuasion techniques is repackage+entice, making up over 48% of the observations.

Table 2: Popularity of SE techniques for tricking the user.

Trick	Total	Percentage
Repackage+Entice	972	48.5%
Invent+Impersonate+Alarm	434	21.7%
Invent+Impersonate+Comply	384	19.2%
Repackage+Decoy	155	7.7%
Impersonate+Decoy	46	2.3%
Impersonate+Entice+Decoy	12	0.6%
Invent+Comply	4	0.2%
Impersonate+Alarm	1	0.1%

Ad-based SE Download Delivery Paths

As shown in Table 1, the majority of SE attacks we observed use online

ads to attract users'attention. To better understand these attacks, we examine the characteristics of their ad delivery path and define the set of nodes (HTTP transactions) on the web path beginning at the first ad node and ending at the download node as the ad delivery path.

Table 3 shows the top 5 "entry point" domain names on the ad delivery paths. Besides performing an analysis of the "entry point" to the ad delivery path, we also analyze the last node on the path, namely the HTTP transaction that delivers the download. Table 4 shows the most popular SE download domains for the comply, alarm and entice classes.

Table 3: Top five ad entry point domains.

Comply	Alarm	Entice
26% onclickads.net	16% adcash.com	20% doubleclick.net
10% adcash.com	7% onclickads.net	16% google.com
10% popads.net	7% msn.com	12% googleadservices.com
7% putlocker.is	6% yesadsrv.com	11% msn.com
3% allmyvideos.net	4% yu0123456.com	8% coupons.com

Table 4: Top five ad-driven SE download domains.

Comply	Alarm	Entice
17% softwaare.net	7% downloaddabs.com	41% imgfarm.com
5% newthplugin.com	4% downloaddado.com	17% coupons.com
5% greatsoftfree.com	4% whensoftisupdated.net	11% shopathome.com
4% soft-dld.com	3% safesystemupgrade.org	5% crusharcade.com
3% younplugin.com	3% onlinelivevideo.org	3% ilivid.com

Ad-Driven Benign Downloads

As a result, one may think that if software is downloaded after clicking on an ad, that software is unlikely to be benign. Somewhat surprisingly, we found that this may not necessarily be the case.

To automatically identify ad-driven benign software downloads, we first derive a set of ad detection regular expressions from the rules used by the popular AdBlock Plus browser extension. We match these regular expressions against the nodes on the reconstructed download path for each benign download.

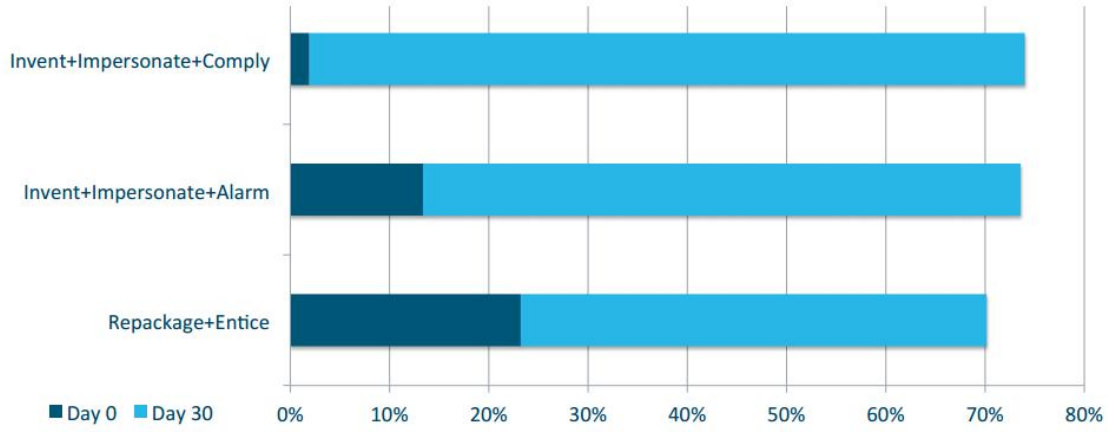
We find that around 7% of all benign software downloads are ad-driven and if a software download is reached by clicking on an ad there is a 40% chance that that software is benign.

4. Detecting SE Download Attacks

We focus on detecting ad-based SE download attacks. We devise a number of statistical features that can be extracted from the network properties of ad-driven software download events, and show that they allow us to build an accurate SE attack classifer, which could be used to detect and stop SE download attacks before they infect their victims.

Antivirus Detection

We scan each SE download sample in our dataset with more than 40 AV engines (using VirusTotal.com). We scanned each of the samples on the day we collected them. Then, we also “aged” the samples for a period of one month, and rescanned them with the same set of AV engines.



SE Detection Classifier

We devise a set of statistical features that can be used to accurately detect ad-driven SE downloads. We focus on ad-driven attacks because they are responsible for more than 80% of all SE downloads we observed.

Given an executable file download event observed on the network, we first automatically reconstruct its download path. Then, translate the download path of each event into a vector of statistical features. Finally, we use the obtained labeled dataset of feature vectors to train a statistical classifier using the Random Forest algorithm that can detect future ad-driven SE download attacks.

The set of detection features:

- Ad-Driven;
- Minimum Ad Domain Age;
- Maximum Ad Domain Popularity;
- Download Domain Age;
- Download Domain Alexa Rank;

Evaluating the SE Detection Classifier

To measure the effectiveness of the SE classifier, we use two separate datasets. The first dataset which we use to train the classifier, includes 1,556 SE download paths, and 11,655 benign download paths. The second dataset consists of new executable downloads that we collected from the same deployment network after the feature engineering phase and after building our detection classifier, contains 1,338 ad-driven SE downloads, and 9,760 benign downloads paths.

After training our SE detection classifier using dataset D1 and the Random Forest learning algorithm, we test the classifier on dataset D2. Table 5 reports the confusion matrix for the classification results. The classifier correctly identified over 91% of the ad-driven SE downloads. Furthermore, it has a very low false positive rate of 0.5%.

Table 5: Confusion matrix for the SE detection classifier.

	Predicted Class	
	Benign	Ad-Based SE
Benign	99.5%	0.5%
Ad-Based SE	8.8%	91.2%

五、相关论文对比

1. An overview of social engineering malware: Trends, tactics, and implications.

We drew from multiple data sources to develop a dataset of social engineering malware incidents and their characteristics. Our primary data source is an online journal, Virus Bulletin, which provides independent advice on developments in viruses and anti-virus products. This journal also serves as a repository of reported malware incidents worldwide. By examining the descriptions of these malware, we discovered that all of them relied on social engineering to be activated and were spread.

Top Malware, 2009.

Position	Name of Malware	Date of Discovery
1	Netsky	February 16, 2004
2	Mytob	April 1, 2005
3	Bagle	January 18, 2004
4	Agent	May 19, 2008
5	Bifrose/Pakes	July 21, 2005
6	Small	August 21, 2008
7	Mywife/Nyxem	September 21, 2005
8	Mydoom	January 26, 2004
9	Zafi	September 14, 2004
10	LovGate	March 27, 2003

Source: *Virus Bulletin* (as of September 15, 2009).

The reported number of security incidents using social engineering malware from 2000 to 2007 shows a definite upward trend (see Fig. 1). We fit an exponential trend model to this data. The analysis shows that

social engineering malware is growing explosively and will continue to pose a substantial security hazard.

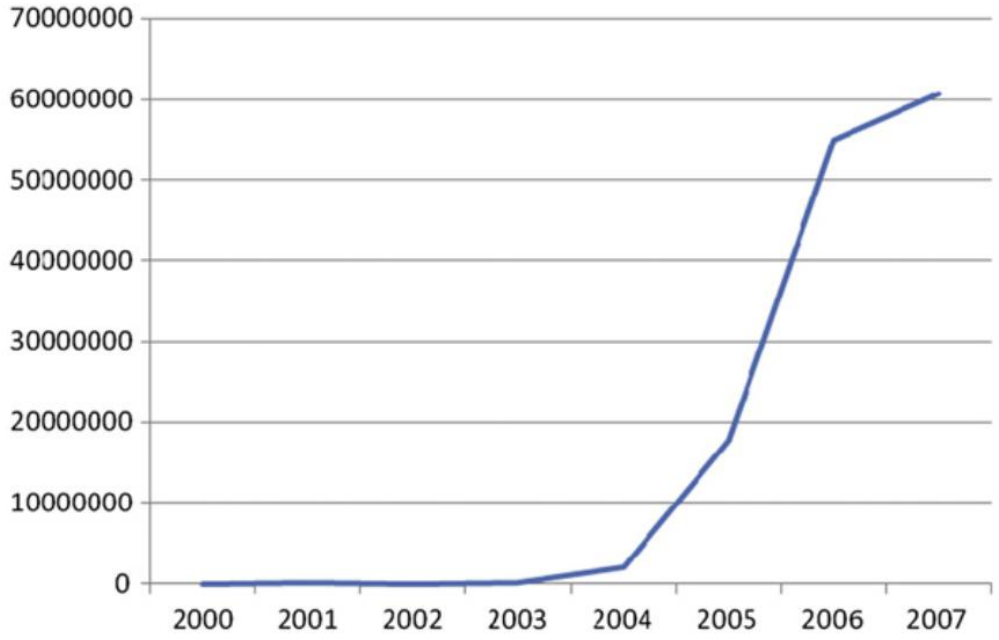


Fig 1. Reported security incidents that used social engineering techniques.

Our analysis of the malware behavior showed some commonalities in steps malware takes to be successfully activated. Using technical details for our sample of malware, we created a master list of actions that alware undertakes to carry out its tasks. The malware's actions generally fell into four categories: (1) persuade a user to activate it, (2) subvert protective technologies, (3) accomplish its mission, and finally (4) propagate. Fig. 2 provides a framework that describes the steps that a social engineering malware undertakes in order to infiltrate a system successfully.

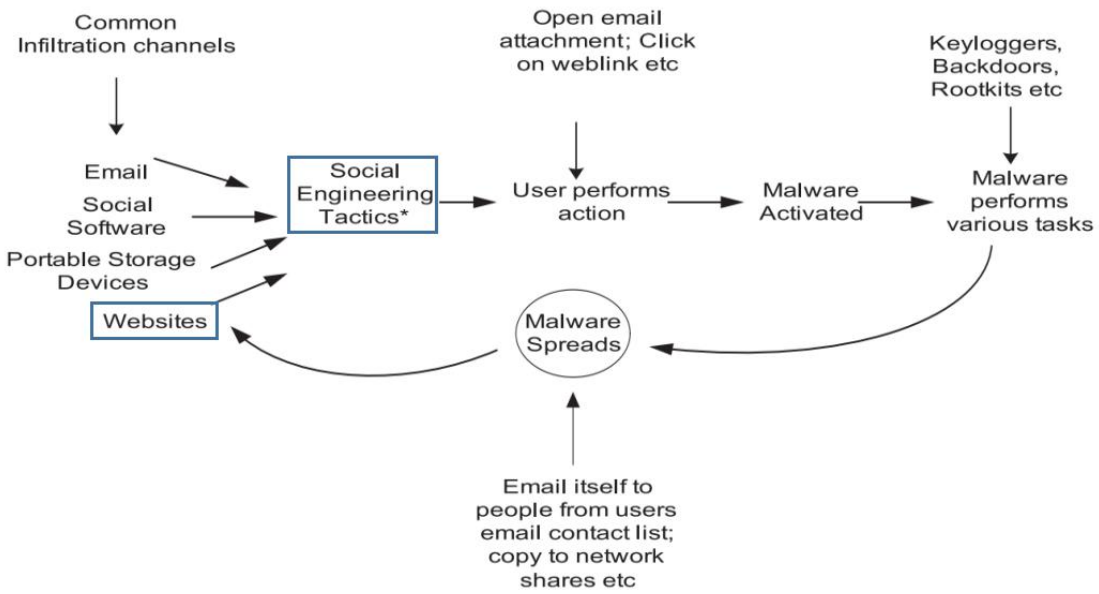


Fig. 2. Steps taken by malware to infiltrate a system.

Social engineering works by manipulating emotions such as fear, curiosity, excitement, empathy, and greed or through the exploitation of cognitive biases. Some persuasive tactics that beguile users are described below.

- Curiosity, empathy, and excitement;
- Fear;
- Greed;

Attackers are now increasing the sophistication of their attack strategies by researching their target base and devising tactics that are tailored to certain user groups. Some examples of user groups targeted by social engineering attackers are described below.

- College students;
- Corporate executives;
- Countries, religious groups, and sects;

By analyzing the technical descriptions compiled from our malware sample, the discussion below provides a general understanding of technical countermeasures that such malware are designed to thwart.

- Combating existing protection mechanisms;
- Fast-Flux;
- Deactivating security tools;
- Own SMTP engine;
- Self-defending;

2. Webwitness: Investigating, categorizing, and mitigating malware download paths.

In this paper, we study the web paths followed by users who eventually fall victim to different types of malware downloads.To this end, we propose a novel incident investigation system, named WebWitness. Our system targets two main goals:

1) automatically trace back and label the sequence of events preceding malware downloads, to highlight how users reach attack pages on the web;

2) leverage these automatically labeled in-the-wild malware download paths to better understand current attack trends, and to develop more

effective defenses.

We build a transactions graph, where nodes are HTTP transactions within the download trace, and edges connect transaction according to a probable source of relationship. Then, starting from the node related to the malware file download, we walk back along the most probable edges until we find a node with no predecessor, which we label as the origin of the download path.

Let D be the dataset of HTTP traffic generated by host A before (and including) the download event. We start by considering all HTTP transactions in D, and construct a weighted directed graph G = (V,E). The vertices are A's HTTP transactions and the edges represent the relation probable source of for pairs of HTTP transactions. Each edge has a weight that expresses the level of confidence we have on the link between two nodes. To build the graph G and draw its edges, we leverage the seven features, an edge e = (v1 → v2) is created if any of the seven features is satisfied. We associate a weight to each of the seven features; the stronger the feature, the higher its weight. Once G has been built, we use a greedy algorithm to construct an approximate backtrace path. The example graph in Fig. 3.

We define the features for automated download path traceback:

- Location ($w_e = 7$)
- Referrer ($w_e = 6$)

- Domain-in-URL ($w_e = 5$)
- URL-in-Content ($w_e = 4$)
- Same-Domain ($w_e = 3$)
- Commonly Exploitable Content ($w_e = 2$)
- Ad-to-Ad ($w_e = 1$)

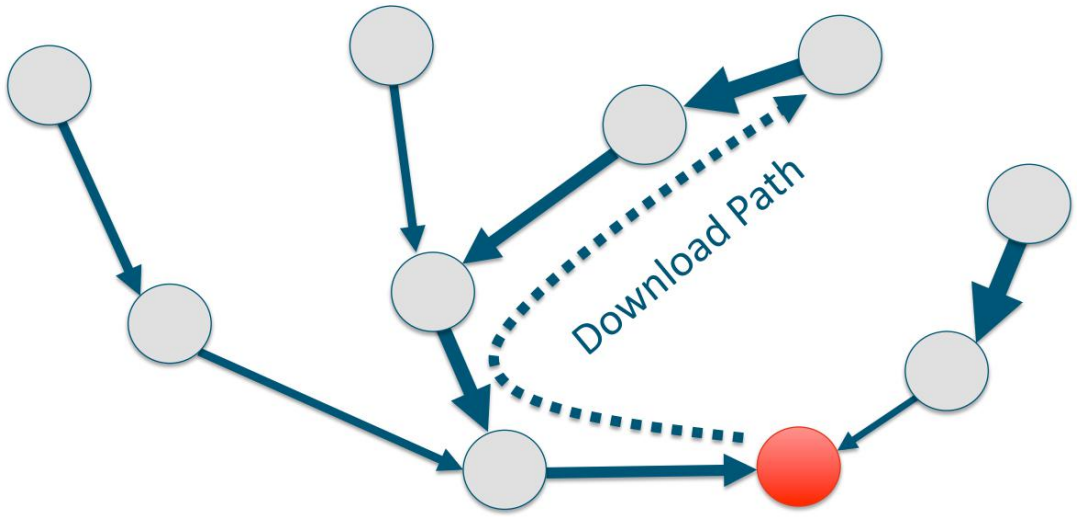


Fig. 3: Example of download path traceback.

3. Camp: Content-agnostic malware protection.

This paper presents CAMP, a content-agnostic malware protection system based on binary reputation that is designed to address the gap between whitelist and blacklist detection for web malware. CAMP is built into the browser and determines the reputation of most downloads locally, relying on server-side reputation data only when a local decision cannot be made.

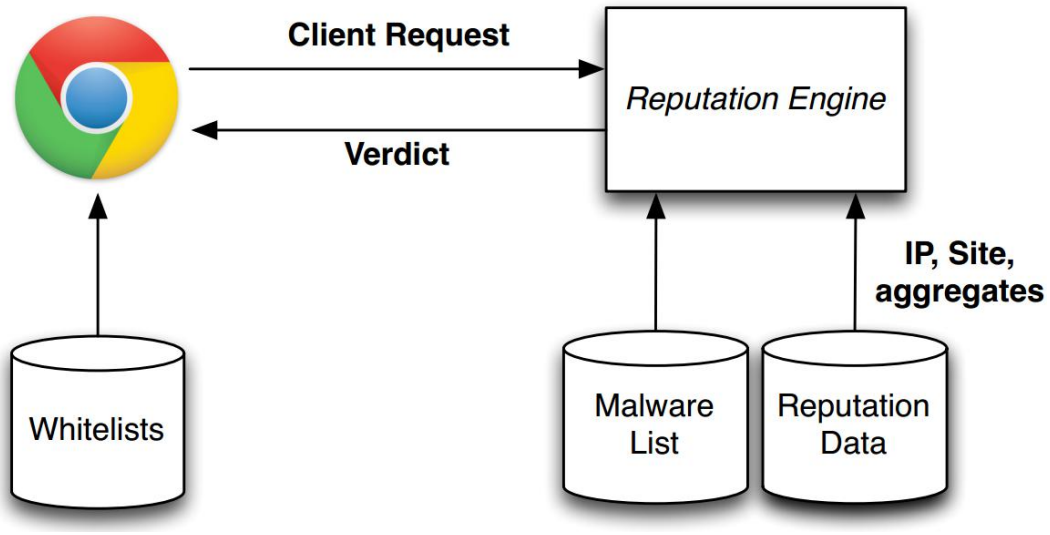


Fig. 4: System Architecture

Client

When a user initiates a binary download in her web browser, the web browser applies several checks before asking the server for a reputation decision. If all the above checks do not result in a local decision, the browser extracts features from the downloaded binary. Since the reputation decision is made on the server, the client can provide as many features as are available to it. The following is a list of features usually available to web browsers:

- The final download URL and IP address corresponding to the server hosting the download.
- Any referrer URL and corresponding IP address encountered when initiating the download, e.g. the results of multiple HTTP redirects.
- The size of the download as well as content hashes, e.g. SHA-256.
- The signature attached to the download which includes the signer as

well any certificate chain leading to it. The client also informs the server if it could successfully verify the signature and if it trusted the signature, e.g. it was rooted in a trusted certificate authority.

Server

The server pipeline has two different roles when processing requests. First, the server receives the client request and renders a reputation verdict based on its reputation system. Second, the server uses the information provided by the client to update its reputation data.