UNIVERSITY OF DEBRECEN
HUNGARY

DETECTING AND COMBATING MALWARE

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science
In

Computer Science and Information Technology

By

AZUAMA KINGSLEY EZECHI
kingsdebrecen@gmail.com

June 2011

SUPERVISED

By

Dr. Fazekas Gábor
(Ph.D., Egyetemi Docens)

Instructor: Dr. Kuki Attila

ACKNOWLEDGMENTS

I would like to give special thanks to God Almighty for the provisions, wisdom and life He give to me. I would also like to give thanks to the following persons:

Dr. Fazekas Gábor: For guiding and supervising my Thesis. I would also like to show appreciation to **Dr. Rachel Marshall**, my Thesis adviser, for his great guidance and support.

Kollár Lajos, Dr. Rutkovszky Edene, Dr. Kuki Attila, Dr. Zichar Mariana, Dr. Baran Sandor and Dr. Sztrik Janos: For their endless help and support during my studies. You are the ones who encouraged and inspired me.

Finally, a special thanks to my friends and family to whom I am deeply grateful for always being there for me.

ABSTRACT

This master thesis observes a process improvement on detecting Malware, although new methods for combating malware have been developed, it is still difficult to communicate and share useful information garnered through these techniques without ambiguity and corresponding data loss. To end this significant gap in malware-oriented communication, this thesis introduces and defines a language for descriptive malware based on its behaviors, artifacts, and attack patterns.

Table of Contents

Acknowledgment i

Abstract of the Master Thesis.. ii

List of Tables iii

List of Figures x

1 Introduction 1

- Addressing standard for descriptive malware.. 2
- A method of malware scanning that evaluates patterns.. 3

2 Background.. 4

- 2.1 Malware Identification & Naming.. 4
- 2.2 CME.. 4
- 2.3 Malware 2.0... 5
- 2.4 How Does Malware Get out of a Computer? 5
- 2.5 What is Malware capable of? 5
- 2.6 Is Malware Infection Likely? 6
- 2.7 How Can You Reduce the Risk of Malware Infection? 6
- 2.8 CIPAV: An Example of Malware Use for Law Enforcement................................ 7
- 2.9 The Dark Side of the Internet.. 7
- 2.9.1 Online theft.. 10
- 2.9.2 Online scams and frauds.. 11
- 2.9.3 Physical harm to people... 12
- 2.9.4 Bullying... 13
- 2.9.5 Defamation and invasion of privacy.. 13
- 2.9.6 Aiding crime.. 14
- 2.9.7 Illegal online gambling.. 14
- 2.9.8 Other pervasive reprehensible behaviors.. 14
- 2.9.9 Causes of the Malware in Internet.. 15
- 2.10 Malware Risk Assessments... 16
- 2.10.1 About Wedge Networks Inc.. 18
- 2.10.2 About Wedge BeSecure.. 18

- 2.10.3 About The Tolly Group 18
- 2.11 PC Tools' new advanced research team in Boulder 18
- 2.12 DHS/DoD/NIST Malware Working Group 20.
- 2.13 Descriptive And Detecting Malware 20

3 Technology-Centric Dark Side of Internet and Objectives 22

- Technology-Centric Dark side of Internet 22

- 3.1 Spam 23
- 3.1.1 E-mail spam 23
- 3.1.1.1 How it works 23
- 3.1.1.2 Anti-spam techniques 24
- 3.1.2 Social spam 27
- 3.1.2.1 How it works 27
- 3.1.2.2 Anti-social-spam techniques 28
- 3.2 Malware 28
- 3.2.1 How they work 29
- 3.2.1.1 Viruses 29
- 3.2.1.2 Worms 30
- 3.2.1.3 Trojans 30
- 3.2.1.4 Spyware 31
- 3.2.1.5 Adware. … 31
- 3.2.2 Anti-malware products Virus scan programs find viruses in one of two ways… 31
- 3.3 Hacking 34
- 3.3.1 How hacking is done 34
- 3.3.1.1 Exploitation of weak authentication 34
- 3.3.1.2 Exploitation of software flaws 35
- 3.3.2 Combating hacking 36
- 3.4 Denial of service attacks 39
- 3.4.1 How DoS works 39
- 3.4.1.1 Ping flood 39
- 3.4.1.2 Smurf attack 40
- 3.4.1.3 UDP flood 40
- 3.4.1.4 TCP SYN flood 40
- 3.4.1.5 DNS amplification attack 40
- 3.4.1.6 Application-level attack 41
- 3.4.1.7 Mail bomb attack 41
- 3.4.1.8 Peer-to-peer attack 41
- 3.4.1.9 Variable-rate and low-rate attacks 41
- 3.4.1.10 Permanent DoS attack (phlashing) 42
- 3.4.2 Preventive measures 42
- 3.4.2.1 Removing known vulnerabilities in protocol behaviors & host configurations… 42
- 3.4.2.2 Filtering the traffic 43
- 3.4.2.3 Detecting attacks 43
- 3.5 Phishing 44

- 3.5.1 How it works.. 45
- 3.5.2 Preventive measures.. 46
- 3.6 Click fraud.. 48
- 3.6.1 How it works.. 48
- 3.6.2 Forms of click fraud and technical responses.. 49
- 3.7 Violation of digital property rights.. 52
- 3.7.1 How DRM works.. 53
- 3.7.2 DRM technology.. 53
- 3.7.3 DRM controversy.. 54
- 3.8 Non-technology-centric responses to the dark side.. 55
- 3.8.1 Legislation.. 55
- 3.8.2 Law enforcement.. 56
- 3.8.3 Litigation.. 58
- 3.8.4 International collaboration.. 58
- 3.8.5 Volunteers actions.. 59
- 3.8.6 Education.. 61
- 3.8.7 Awareness and caution.. 62
- 3.9 Testing of anti-rootkit software for the detection and removal of rootkits iii...... 62
- 3.9.1 Combating Malicious Software.. 63
- 3.9.2 Malware sample sources for Researchers.. 63
- 3.9.3 Analyzing Malicious Documents Cheat Sheet.. 64
- 3.9.4 Updates from Twitter Users Who Cover Malware.. 64
- 3.9.5 On-Line Tools for Malicious Website Lookups.. 64
- 3.9.6 Block lists of Suspected Malicious Ips and URLs….. 64
- 3.9.7 Automated Malware Analysis Services.. 65
- 3.9.8 Reverse-Engineering Cheat Sheat.. 65
- 3.9.9 Using VMware for Malware Analysis.. 65
- 3.9.11 The Evolution of Malicious Agents.. 65
- 3.9.12 Stopping Malware on its Tracks.. 65
- 3.9.13 Be attuned to the state of your network and systems.. 66
- 3.9.14 Trap malware with honeypots.. 66
- 3.9.15 Protect the endpoint from malware threats.. 67
- 3.9.16 A comprehensive security program is a must.. 68
- 3.9.17 National Security Letters.. 69
- 3.9.18 Foreign Intelligence and Terrorism Investigations.. 70
- 3.9.19 Defensive Technology.. 70
- 3.9.20 Information Stored By Third Parties.. 72
- 3.9.21 Data Stored on Your Computer.. 72

- Objectives.. 72

- 3.10. Non-signature Malware Description.. 73
- 3.10.1 Malware Grouping.. 73
- 3.10.2 Blended Malware Threat Description.. 73
- 3.10.3 Mitigation of Armoring Techniques.. 73

3.11 Unambiguous Malware Typing ... 73
3.12 Improved Malware Pedigree Tracking ... 74
3.13 Legal Malware Definitions ... 74

4 Scope and Proposed Framework ... 76

Scope ... 76
Proposed Framework ... 76
4.1 DADM's Enumerated Attributes ... 77
4.1.1 Low-level Attributes ... 78
4.1.2 Mid-level Behaviors ... 79
4.1.3 High-level Taxonomy ... 79
4.1.4 Metadata ... 79
4.2 The DADM Schema ... 80
4.2.1 Namespaces ... 81
4.2.2 Properties ... 81
4.2.3 Relationships ... 81
4.3 The DADM Cluster ... 81
4.3.1 Output format ... 83

5 Description Test Case ... 84

5.1 Initial Description ... 84
5.2 Second Description ... 85
5.2.1 General Observations ... 86
5.2.1.1 Attributes ... 86
5.2.1.2 Schema ... 87
5.2.2 Open Questions ... 87
5.2.2.1 Attributes ... 87
5.2.2.2 Schema ... 88

6 Ties to MSM Standards ... 89

6.1 CAPEC ... 90
6.2 CEE ... 90
6.3 CWE ... 90
6.4 CVE ... 90
6.5 CPE ... 91
6.6 CCE ... 91
6.7 OVAL ... 91

7 Use Cases ... 92

7.1 12 Analysis Use Cases ... 92

- 7.1.1 Behavioral/Dynamic Analysis Tool Wrapper 93
- 7.1.2 Malware Repositories 94
- 7.1.2.1 Improved Data-mining 95
- 7.1.3 Objective Criteria for Tool Assessments 96
- 7.2 Operational Use Cases 96
- 7.2.1 Uniform Malware Reporting Format 97
- 7.2.1.1 Multiple Source Integration 98
- 7.2.2 Malware Detection 98
- 7.2.2.1 Host-based Detection 99
- 7.2.2.2 Network-based Detection 99
- 7.2.3 Malware Threat Assessment 100
- 7.2.4 Malware Response 101
- 7.2.4.1 Remediation 101
- 7.2.4.2 Countermeasures 102
- 7.2.5 Linking Malware TTPs 102

8 Conclusions 103

- 8.1 Implementation Issues 103
- 8.1.1 DADM Schema Structuring 103
- 8.1.2 Temporal/Chronological Characterization 103
- 8.1.3 Static Data-set Incorporation 104
- 8.1.4 Assembly Code Characterization 104
- 8.1.5 Malware Metadata 104
- 8.2 General Challenges 105
- 8.2.1 Repeatability 105

References 106

List of Figures

Figure 1 E-mail spam-filtering process.. 25

Figure 2 Anti-malware products in the cloud... 33

Figure 3 Use of a canary word for buffer overflow protection.. 37

Figure 4 SYN cookie... 42

Figure 5 Ingress and egress filtering at the edge routers.. 44

Figure 6 Steps in mutual authentication..48

Figure 7 Bloom filter for detecting duplicate clicks... 49

Figure 8 Click fraud using paired pages..51

Figure 9 Repeatedly obtaining new IP addresses to a computer... 51

Figure 10 (a and b) Example coalition attack... 52

Figure 11 DADM Overview.. 77

Figure 12 DADM's Tiers of Enumerated Elements.. 78

Figure 13 Example of Structure Imparted through DADM's Schema……………………… 80

Figure 14 DADM Cluster Overview.. 82

Figure 15 Notional DADM Cluster – XML Representation... 83

Figure 16 Conficker.B File Sharing Propagation Mechanism.. 86

Figure 17 DADM's Relationship with MSM Standards.. 89

Figure 18 Typical Analysis DADM Usage Scenario... 92

Figure 19 Multiple Tool Output to Common Format.. 93

Figure 20 Repository Data Sharing Through DADM.. 94

Figure 21 DADM for use in Operations Security Management.. 96

Figure 22 DADM as a Uniform Malware Reporting Format... 97

Figure 23 Malware Detection via Shared Attributes... 98

CHAPTER 1

INTRODUCTION

Malware has been around in one form or another since 1971 (the advent of the first PC virus). In its various forms, from spyware to rootkits, it is presently responsible for a host of illicit activities, ranging from the vast majority of spam email distribution through botnets [6], to the theft of sensitive information via targeted social engineering attacks [30]. Effectively an autonomous agent operating on behalf of the attacker, malware has the ability to perform any action capable of being expressed in code and represents a prodigious threat to cyber security.

The protection of computer systems from malware is therefore currently one of the most important information security concerns for organizations and individuals, since even a single malware infection can result in damaged systems and compromised data. Being disconnected from a computer network does not completely mitigate this risk of infection, as exemplified by the recent wave of malware that utilizes USB as a vector [31]. As such, the main focus of a large number of anti-malware efforts to data has been on preventing damaging effects through early detection.

There are currently several common methods utilized for malware detection, based mainly on physical signatures and heuristics. These methods are effective in terms of their narrow scope, although they have their own individual drawbacks, such as the fact that signatures do not scale and are therefore unsuitable for dealing with zero-day, targeted, polymorphic, and other forms of emerging malware. Similarly, heuristic detection may be able to generically detect certain types of malware while missing those that it does not have patterns defined for, such as kernel-level rootkits. These methods, while still useful, cannot be exclusively relied upon to deal with the current influx of malware.

More modern methods for detecting and combating malware often rely on the characterization of malware attributes and behaviors [11-14]. Such behaviors and attributes are commonly discovered through the use of static [15] and dynamic [16] analysis techniques. The combination of the two allows for an encompassing profile of malware to be constructed based upon its dis-assembly and observed run-time behavior. Yet, such techniques are hampered by the non-existence of a widely accepted standard for unambiguously characterizing malware.

The lack of such a standard means that there is no clear method for communicating the specific malware attributes detected in malware by the aforementioned analyses, nor for enumerating its fundamental composition. Several major problems result from this, including non-interoperable and disparate malware reporting between organizations, disjointed or inaccurate malware attribution, the duplication of malware analysis efforts, increased difficulty in determining the severity of a malware threat, and a greater period of time between malware infection and detection/response, among others.

Information security is the protection of information from a wide range of threats in order to ensure business continuity, minimize business risk, and maximize return of investments and business opportunities. Identifying and classifying threats to information systems is vital to building defensive mechanisms. As organizations become vigilant about protecting their networks by investing in better security technologies, attackers have focused their attention on exploiting the weakest link in security – end users. Human error is often a major cause of problems in technological implementations, and people are generally considered the weakest link in an information security program.

A major threat to organizational information security is the rising number of incidents caused by social engineering attacks. Social engineering is defined as the use of social disguises, cultural ploys, and psychological tricks to get computer users to assist hackers in their illegal intrusion or use of computer systems and networks. Social engineering is one of the strongest weapons in the armory of hackers and malicious code writers, as it is much easier to trick someone into giving his or her password for a system than to spend the effort to hack in.

Despite the continued effort of organizations to improve user awareness about information security, social engineering malware has been successful in spreading across the Internet and infecting numerous computers. By 2007 social engineering techniques became the number-one method used by insiders to commit e-crimes, but unsuspecting users remain the predominant conduit for the authors of malicious code. Given this susceptibility to social engineering techniques, our primary research objective is to identify and describe social engineering malware trends and tactics.

For social engineering malware to be successful, it needs to be activated by the end user and run on the system without interruption. But if the malware can be prevented from reaching users it will not be successful. Identifying attack strategies is vital to developing countermeasures that can be incorporated into preventive mechanisms like e-mail filtering and end-user security training. Information on the behavior of the malware during propagation helps in the creation of early warning systems. Even if the malware bypasses the prevention stage and is executed by a user, if it can be detected on the machine and blocked from performing its harmful routine, the effects of the malware can be alleviated.

When computer malware is activated, it makes various changes in the computer by opening back doors that enable it to spread to other machines. It also executes defensive strategies in order to remain undetected. Identification of such defensive strategies is helpful in discovering malware in its early stages on end-user machines, and blocking it from being executed completely and propagating further. Identifying it also aids in the development of heuristics analysis – a method of malware scanning that evaluates patterns of behavior to discover anomalies. Security researchers use this data to build behavioral models of malware, and end users can receive alerts notifying them of unusual behavior on their machines.

In order to provide guidelines for strengthening an organization’s defenses against social engineering malware, we gathered data on such malware and analyzed their characteristics. The analysis allowed us to identify the strategies employed, both psychological and technical. This book provides empirical evidence of the growing reach of social engineering malware. The following section describes our data collection process and summarizes trends in social engineering malware incidents.

Next we present a framework for the propagation of social engineering malware and discuss some common avenues of attack. This discussion is followed by an analysis of the psychological tactics used by the malware as well as descriptions of some of the technical features that are designed to help the malware counter existing security precautions. The book concludes with a discussion of the implications of our findings as well as recommendations to mitigate the threat posed by social engineering malware.

On this basis, it is clear that a standard for descriptive malware in terms of its attack patterns [17], artifacts, and actions is needed to address such issues and allow for the clear communication of the information gained using static and dynamic analysis. This is the approach being taken by the Descriptive And Detecting Malware (DADM) effort. The characterization of malware using such abstract patterns offers a wide range of benefits over the usage of physical signatures. Namely, it allows for the accurate encoding of how malware operates and the specific actions that it performs. Such information can not only be used for malware detection but also for assessing the end-goal the malware is pursuing and the corresponding threat that it represents.

The rest of the chapter's is structured as follows. In Chapter 2, we lay out the history and major factors leading up to this book. Chapter 3, details of Technology-centric dark side of Internet and describes our objectives for this project. Chapter4 briefly covers the scope of our proposed language & discuss the high-level details of the DADM framework. Chapter 5 deals with description test case for DADM. Chapter 6 covers the Ties to MSM standards and relationships. Chapter 7, details of use cases. Chapter 8, finally conclusions of open issues and challenges fasting Descriptive And Detecting Malware (DADM).

A glossary of commonly used terms can be found at the end of the document in Appendix A (with terms being used for the first time highlighted in italics).

CHAPTER 2

BACKGROUND

2.1 MALWARE IDENTIFICATION & NAMING

Viruses and other malware are commonly identified by AV product vendors and others using the Computer Anti-virus Researcher Organization (CARO) naming scheme [19], first adopted in 1991. Although the value of knowing the specific malware threat that one is dealing with is significant, the fact remains that the CARO naming scheme is not an official standard and is not applied consistently among its adopters. As a result, it is often the case that a single malware instance has multiple, disparate CARO-based identifiers1 thus furthering confusion regarding malware identity and subsequently reducing the value of the identifier.

A large part of the problem with a system like the one created by CARO is that it attempts to encode malware attributes as part of the identifier [20]. It is very difficult to include all of the important information regarding a malware instance inside its identifier without making the identifier too large and cumbersome to effectively utilize. Likewise, there is no way of determining which critical attributes should be present in the identifier and which to leave out, as certain attributes are relevant only to certain parties. It appeared that a better solution would be the use of standardized, non-attribute based malware identifiers.

2.2 CME

In the fall of 2004, MITRE began work on the Common Malware Enumeration effort. The goal of CME was to provide single, common identifiers for new and prevalent virus threats, in order to reduce public confusion during malware incidents. Like other MITRE security standards efforts, the intent was to collaborate with industry and develop a consensus-oriented approach. This community effort was not an attempt to replace the vendor names used for viruses and other forms of malware, but rather was intended to facilitate a shared, neutral indexing capability for malware.

In the first quarter of 2005 an initial capability was stood up. Anti-virus vendors submitted malware samples to a submission server with some metadata, the CME Board (composed of AV researchers and MITRE engineers) decided when a CME identifier should be generated, and the CME team identified the mapping between this sample and vendor names and created the content for the CME web site. For a period of time, CME seemed to fulfill its purpose, with identifiers being issued for prevalent malware incidents, and likewise being referenced in popular reporting about such malware [22]

2.3 MALWARE 2.0

Malware is a catch-all term referring to software that runs on a computer and operates against the interests of the computer's owner. , worms, Computer viruses, "spyware", rootkits and key logger are often cited as subcategories of malware. Note that some programs may belong to more than one of those categories.

2.4 HOW DOES MALWARE GET ONTO A COMPUTER?

Some malware is spread by exploiting vulnerabilities in operating systems or application software. These vulnerabilities are design or programming errors in software that can allow a clever programmer to trick the defective software into giving someone else control. Unfortunately, such vulnerabilities have been found in a wide variety of mainstream software, and more are detected all the time — both by those trying to fix the vulnerabilities and by those trying to exploit them.

Another common vector by which malware spreads is to trick the computer user into running a software program that does something the user wouldn't have wanted. Tricking the user is a pretty powerful way to take over a computer, because the attacker doesn't have to depend on finding a serious weakness in mainstream software. It is especially difficult to be sure that computers shared by several users, or a computer in a public place such as a library or Internet café, are not compromised. If a single user is tricked into running a malware installer, every subsequent user, no matter how cautious, could be at risk. Malware written by sophisticated programmers generally leaves no immediately visible signs of its presence.

2.5 WHAT IS MALWARE CAPABLE OF?

Malware is extremely bad news from a security and privacy perspective. Malware may be capable of stealing account details and passwords, reading the documents on a computer (including encrypted documents, if the user has typed in the password), defeating attempts to access the Internet anonymously, taking screen shots of your desktop, and hiding itself from other programs. Malware is even capable of using your computer's microphone, web-cam, or other peripherals against you.

- I) The chief limitation in malware's capability is that the author needs to: have anticipated the need for the malware to do something,
- ii) Spent a substantial amount of effort programming the malicious feature, testing that it works and is robust on numerous different versions of an operating system, and
- iii) Be free of legal or other restrictions preventing the implementation of the feature.

Unfortunately, a black market has appeared in recent years that sell malware customized for various purposes. This has reduced the obstacles listed in category (ii) above.

The most alarming feature of malware is that, once installed, it can potentially nullify the benefits of other security precautions. For example, malware can be used to bypass the protections of encryption software even if this software is otherwise used properly. On the other hand, the majority of malware is mainly designed to do other things, like popping up advertisements or hijacking a computer to send spam.

2.6 IS MALWARE INFECTION LIKELY?

Nobody knows how many computers are infected with malware, but informed estimates range from 40% to almost 90% of computers running Windows operating systems. Infection rates are lower for MacOS and Linux systems, but this is not necessarily because Windows is an easier target. Indeed, recent versions of Windows are much improved in security. Rather, more malware authors target Windows machines because an effective attack will give them control of more computers.

The risk that any given computer is infected with malware is therefore quite high unless skilled computer security specialists are putting a substantial amount of effort into securing the system. With time, any machine on which security updates are not installed promptly is virtually guaranteed to become infected. It is however overwhelmingly likely that the malware in question will be working on obtaining credit card numbers, obtaining eBay account passwords, obtaining online banking passwords, sending spam, or launching denial of service attacks, rather than spying on specific individuals or organizations.

Infection by malware run by U.S. law enforcement or other governmental agencies is also possible, though vastly less likely. There have been a handful of cases in which it is known that warrants were obtained to install malware to identify a suspect or record their communications (see the section on CIPAV below). It is unlikely that U.S. government agencies would use malware except as part of significant and expensive investigations.

2.7 HOW CAN YOU REDUCE THE RISK OF MALWARE INFECTION?

Currently, running a minority operating system significantly diminishes the risk of infection because fewer malware applications have been targeted at these platforms. (The overwhelming majority of existing malware targets only a single particular operating system.)

Vulnerabilities due to software defects are difficult to mitigate. Installing software updates promptly and regularly can ensure that at least known defects are repaired. Not installing (or running)any software of unknown provenance is an important precaution to avoid being tricked into installing malware.

This includes, for example, software applications advertised by banner ads or pop-ups, or distributed by e-mail (even if disguised as something other than a computer program).

Recent operating systems attempt to warn users about running software from an unknown source; these security warnings serve an important purpose and should not be casually ignored. Strictly limiting the number of users of a computer containing sensitive information can also be helpful. Notably, some malware targets children, including malicious code along with downloadable video games. (Of course, computer users of any age can be tricked into installing malware!)

On Windows, regularly running antivirus and antispyware software can remove a large proportion of common malware. However, this software is not effective against all malware, and must be regularly updated. Since anti-malware software is created by researching malware discovered "in the wild," it's also probably ineffective against uncommon, specially-targeted malware applications that aim to infect only a few specific computers rather than a large population on the Internet.

2.8 CIPAV: AN EXAMPLE OF MALWARE USE FOR LAW ENFORCEMENT

A CIPAV is an FBI acronym which stands for Computer and Internet Protocol Address Verifier. CIPAVs are a type of malware intended to identify people who are hiding their identity using proxy servers, bot nets, compromised computers or anonymity networks like Tor. A small amount is known about them as a result of published documents from cases in which they were used. CIPAVs may include use of browser exploits to run software on a computer regardless of how many steps of indirection are present between the attacking server and the user.

2.9 THE DARK SIDE OF THE INTERNET

During the past 15 years, the Internet and the Web have dramatically transformed the world. Today we live in the initial stage of a revolution in the history of mankind. The Good of the Internet are all the benefits of the Internet that people enjoy. These include the ability to instantly access all kinds of information anytime from anywhere; transformation and expansion of business, governance, education, research, entertainment, culture, etc. ; mobile life and mobile work; online social communities that have given people opportunities to express themselves and reach out to many people they had never met offline; etc.

The Bad are all the bad things that annoy and damage people, businesses, institutions, and governments. It is the result of illegal, unethical or reprehensible behaviors of people, and nobody can escape from this. The Bad include spam, computer viruses and worms, hacking, denial of service attacks, online frauds, identify theft, violation of digital property rights, violation of privacy, online bullying, reprehensible online behaviors, etc.

The Ugly are the things that fall outside the purview of the law, and need time to get sorted out. In the midst of a revolution, the fortunes of individuals, businesses, industries, and even governments can undergo an upheaval. For example, the music, newspaper, entertainment, advertising, software, and communications industries are now undergoing major changes.

Further, some of what should be Bad have become the Ugly, because people's attitudes have changed and the sheer number of law breakers has overwhelmed the governments' law-enforcement capacities.

A case in point is the wide-spread unauthorized posting and sharing of copyrighted digital properties, such as music, video, movies, software, etc In this paper, we explore the dark side of the Internet [39,40], that is, the Bad things the Internet has brought about. The fact that the Internet would come with the dark side should not have been a surprise, for it is inevitable. Both the offline and online worlds are inhabited by the same people.

The dark side of both worlds is the natural consequence of the innate human failings and interactions among people. Normal people have emotions, desire for financial gains, psychological needs, and psychological imperfections. The emotions include enmity, jealousy, anger, insecurity, selfishness, etc. Further, many people are immature and naı¨ve. All elements of the dark side of the Internet are illegal or unethical or, at least, reprehensible. The global reach, speed of dissemination, anonymity, cross-border nature of the Internet, and the lack of appropriate laws or international agreements have made some of them very wide-spread, and very difficult to prosecute.

Nevertheless, the dark side of the Internet has to be dealt with, so that the Internet revolution can continue to change the world for the better. However, just as it has not been possible to rid the offline world of the dark side, it will not be possible to rid the online world of the dark side. The best that can be done for the online, World is to control the dark side to a tolerable level, just as has been the case for the offline world. The dark side of the Internet is the result of the abuse of the Internet and Web technology by many people.

There are various types of responses to mitigate the dark side, including both technology-centric and non-technology-centric ones. Non-technology responses include legislation, law enforcement, litigation, international collaboration, civic actions, education, and awareness and caution by people. Much has been written about some of the elements of the dark side, mostly in the form of Wikipedia articles; blogs; and articles in trade journals, magazines and newspapers.

However, there is a paucity of scholarly work that brings them together. We hope that this paper will become one of the comprehensive starting references on the subject of the dark side of the Internet. We provide a broad coverage of the subject, including the elements of the dark side, the types of damages done by the dark side, the causes of the dark side, and the approaches to mitigating the dark side. To make our discussions rigorous, we provide taxonomies of all of these. In consideration of the publication venue of this paper, we discuss the technology responses in much greater depth.

Hundreds of millions of Internet and Web users have had to bear the burden caused by the transgressions of perhaps hundreds of thousands of people. The damages come in the form of loss of money, defamation, invasion of privacy, and physical harm. They also include loss of time and mental anguish or psychological damage that come from all of above types of damages. Cyber bullying is another main cause of mental anguish. An increase in crime also results from dissemination of crime-aiding information.

The dark side may be divided into two groups: technology centric, and non-technology-centric. We note that the dark side of the Internet has been brought about by Internet technology, including Web browser, Web server, e-mail,posting and downloading of text and multimedia documents, etc. In that sense, all of the dark side is technology-based. Nonetheless, for the purpose of organizing this paper, and as a basis for understanding the dark side of the Internet more clearly, we define technology-centric dark side as those elements of the dark side that require technologies or technical skills beyond the main-stream Internet technology that most people use for their daily work and life, and which require new technologies or enhancements to existing technologies in order to address them.

The technologies and technical skills beyond the main-stream Internet technology include, for example, software to harvest e-mail addresses, techniques to create and disseminate malware, software to manage many computers that are infected with malware, techniques to hack into others' computers (other than simple password guessing or hiring people in India to identify the CAPTCHA code, etc.). Technology-centric dark side includes seven elements: spam, malware, hacking, and denial of service attacks, phishing, click fraud, and violation of digital property rights. We will consider all other elements of the dark side as non-technology-centric.

Non-technology-centric dark side each has an offline counterpart. Further, to address them requires mostly the ability to trace those who sent e-mail or service requests, and those who posted documents to the Internet. Below, we first discuss the seven technology-centric elements of the dark side. we will keep their discussions brief here Of the seven elements, in our view, malware, hacking, denial of service attacks, and click fraud do not have offline counterparts. Junk mail is the offline counterpart to spam; hacking of the CDs and DVDs protected with digital rights management technology predates the Internet; and theft of bank account information and people's identities, too, predate the Internet

- Spam

Spam is the true scourge of the Internet. Some estimate that there are 200 billion spam messages a day [3]. Nobody can escape from spam. Spam refers to unwanted notices. They are mostly marketing notices. Spam comes mostly in the form of e-mail. However, spam also comes in the form of instant messages, text messages, and Internet telephone calls. Some of the spam attempt to scam people or spread malware.

- Malware

Malware include viruses, worms, Trojan horses, spyware, and adware. A virus is a computer program that attaches itself onto a host (e.g., a program file or a hard disk boot record) and spreads when the infected host is moved to a different computer. In August 2010, Postini blocked 188 million viruses [42]. A worm is a computer program that can replicate itself and spread across a network. A Trojan horse is a computer program that appears to be a legitimate program but has malicious code hiding inside which runs when activated. Spyware is malware that collects and sends data copied from the victim's computer, such as financial data, personal data, passwords, etc. Adware, or advertising supported software, is a computer program that automatically displays ads.

- Hacking

Hacking refers to breaking into others' computers. The Internet has opened possibilities for hackers to break in without gaining physical access to their target computers. Some hackers hack just to test or show off their technical skills. Some hack to gain access to sensitive
data, steal money or digital properties, destroy data or cause system break downs.

- Denial of service attacks

Denial of service (DoS) attack floods a target computer system with bogus requests, making it unable to provide normal services to intended users. DoS may use brute force flooding of the target computer system, or exploit flaws in the network or the target system. DoS attack does not involve breaking into the target system.

- Phishing

Phishing refers to luring Internet users to Web sites that masquerade as legitimate Web sites, such as banks, credit card companies, auction sites, popular social Web sites, Internet service sites, etc., and directing the users to enter sensitive information, such as user names, passwords, credit card details, bank account details, etc. The information thus acquired may be re-sold or used to commit cyber crimes, such as theft, identity theft, frauds, etc.

- Click fraud

Click fraud refers to clicking on online ads for the purpose of making money or inflict financial damage to the advertisers.

- People directly violate digital property rights

People directly violate digital property rights of others by posting or disseminating, without authorization, digital properties, such as music, movies, software, books, etc. People indirectly violate, or facilitate violation of, digital property rights by posting or disseminating file-sharing programs for others to use to share digital properties without authorization. People also indirectly violate digital properties by posting or disseminating confidential data, such as decryption keys or activation keys, for the properties.

The non-technology-centric dark side of the Internet may be classified into eight elements. These include online theft, online scams and frauds, physical harm to people, defamation and invasion of privacy by spreading false or private information, illegal online gambling, aiding crimes, and general reprehensible behaviors.

2.9.1. ONLINE THEFT

Online theft includes theft of electronic funds, data, identity, and digital properties. Online theft is committed by first committing data theft. Data theft refers to theft of such data as credit and debit card numbers; user names, passwords, account information, personal profiles, etc. From the computers of individuals, financial institutions, Web sites, etc. Data theft may be committed through various offline means, such as packet sniffing, wire tapping, reading off a memo pad with passwords written on, rummaging through a trash can, etc.

It may also be committed through hacking, phishing, and use of malware. Identity theft refers to theft of detailed personal data, including social security numbers (for US citizens and residents) or resident registration numbers, passport numbers, birth certificate numbers, personal history data, membership data, purchase records, tax records, etc. This is committed largely for financial gains, such as to withdraw bank account balances, charge credit cards, take out loans, etc.

However, it has also been used to facilitate many types of offline crimes; the identity thieves pile up criminal records under the names of the victims. On the Internet, identity thieves acquire people’s personal information not only by using such illegal means as phishing, hacking, and malware, but also by simply browsing the personal profiles and posts on social networking Web sites. They often resell massive amounts of personal information they steal. It is usually difficult and time-consuming for the victims of identity theft to clean up the effects of the identity theft.

Identity Theft Resource Center (http://www.idtheftcenter.org) estimates that on average it takes about 330 h for a victim to clean up the effects. Cleaning up includes nullifying financial liabilities and criminal records piled up by the thieves. The records may be kept by the police, courts, news media, credit bureaus, data aggregators, etc. Identity theft has been on the rise. In 2008, in the United States alone, there were 9.9 million victims [43]. One approach to discover if identity theft may have taken place is to search for identity information in a database of stolen records.

2.9.2. ONLINE SCAMS AND FRAUDS

Online scams and frauds are confidence games that trick people into believing that they will receive money or property, and giving up real money or real properties, and end up receiving nothing of value. Just as in the offline world, there are many types of scams and frauds in the online world [44]. (We note that phishing and click fraud are also online frauds.

However, we classify them as elements of the technology-centric dark side.) Non-technology-centric online scams and frauds include purchase scams, money transfer fraud, dating scam, investment frauds, cramming, etc. Each of these has many variations. In a basic purchase scam, the scammer has a merchant charge (what will turn out to be) a bogus credit card and ship goods. The merchant later gets hit with a charge back from the credit card processor.

The purchase scams have a number of variations: counterfeit money order, automobile, check cashing, reshipping, call tag, business opportunity/ work-at-home. Under a counterfeit money order scam, the fraudster sends a counterfeit money order to a victim, and has the victim purchase goods (using his own real money) and ship them to the fraudster. Under a reshipping scam, the scammer has the victim charge a stolen or fake credit card, and ship goods to a country with a weak legal system. The ‘‘call tag’’ scam is somewhat elaborate.

The scammer, using a stolen credit card, purchases goods online for shipment to the real card holder. The scammer then gets permission from the real card holder to have a shipping company pick up and reship the ‘‘mistakenly shipped’’ goods to the scammer. The real card holder, upon noticing a charge in his card, generates a charge back to the unsuspecting merchant. Under the business opportunity scam, which has several variations, the scammer offers a victim a business opportunity to earn lots of money in a ‘‘work at home’’ venture. The scammer requires the victim to make a payment for materials or instructions needed to start the ‘‘work at home’’ venture, but does not deliver anything, or merely sends advice on how to place ads similar to the one that recruited the victim.

The money transfer fraud lures a victim with an employment offer, and asks him to help transfer money to a foreign company. The fraudster then sends a fake check or postal money order, in the hopes of getting the victim to deposit it and sending real money.

Through repeated interactions with the victim, the fraudster can acquire the victim’s full name, address, social security number, bank account number, etc. (i.e., the victim’s identity). Dating scams take a long time to start bearing fruit. The fraudster develops a relationship online with a victim, and eventually convinces the victim to send money.

Under the investment fraud, fraudsters manipulate securities prices on the market by disseminating false or fraudulent information in chat rooms, forums, Internet boards, and via e-mail (spamming). The investment fraud includes pump and dump, and short sell. In the pump and dump fraud, the fraudsters cause a dramatic price increase in lightly traded stocks or stocks of shell companies (the ”pump”). As soon as the price reaches a certain level, the fraudsters immediately sell off their holdings of those stocks (the ”dump’’). In the short sell fraud, the fraudsters cause a dramatic decrease in a specific company’s stock.

Once the stock reaches a certain low level, the fraudsters buy the stock or options on the stock. The fraudsters “pump’’ the stock price or just wait for the stock price to go back to its normal level. When the price reaches a certain high level, the fraudsters sell the stock. Under the Internet marketing and retail scam, the victim is tricked into giving credit card information or sending cash, in exchange for goods or services. The goods never arrive, turn out to be fakes, or are worth far less than advertised. A common example of this type of scam is pornographic Web sites that advertise free access, but require credit card information “for age verification purposes only’’. The scammer makes large charges to the credit card.

Cramming is a telemarketing scam that puts unauthorized or nonexistent telecommunication or service charges on the consumer’s credit card, bank account or phone bill. The scammer captures a consumer’s telephone number, using automatic number identification, a system similar tocaller ID. Then by luring a consumer to make calls, the scammer can cause charges for a product or service to be included on that person’s phone bill [45].

2.9.3. PHYSICAL HARM TO PEOPLE

There are various ways in which use or misuse of the Internet can lead to physical harm to people. There are suicide Web sites that encourage members to get together and commit suicide together. In South Korea, during the past several years, there have been several reports of two or three people who met on suicide Web sites and committed suicide together. Many people join social networking Web sites to hunt for sex victims. These include pedophiles, child pornographers, sex merchants who lure children into prostitution, etc.,

The number of such people is incredibly high. Upon subpoena from the government, the social networking site MySpace had to turn over a list of members who are convicted sex offenders. The list contained 90,000 members [49]. These are people who used their real names. It is not known how many sex offenders do not use their real names on MySpace. It is feared that many of these people may be on the prowl for preys among the young members of MySpace.

2.9.4. BULLYING

Cyber bullying (or stalking) is defined as hurting or embarrassing another person using the Internet, cell phones or other devices by sending or posting text or images [50]. (Although the term cyber stalking is sometimes defined to mean bullying involving adults, we will not distinguish the two terms, and use the term cyber bullying to refer to cyber stalking, too.) Almost half of all American teenagers have been subjected to cyber bullying, and many adults have been bullied and disparaged on Internet forums, blogs, etc. [50-51].

Using the computer or mobile phones makes it easier for cyber bullies to be cruel and abusive, and to resort to creating and spreading false stories about their victims. Cyber bullies penetrate the walls of the homes of the victims by sending instant messages, text messages, and emails; or by simply posting comments to online forums, blogs, and social networking Web sites. Online bullying causes psychological damages to the victims, including lower self-confidence, depression, fear of people, reluctance to participate in group activities, etc.

It sometimes even leads to suicide. In the United States alone, at least four teenage victims of cyber bullying committed suicide [50]. In particular, in 2006, a woman, Lorie Drew, by masquerading as her daughter on the social networking site MySpace, drove her daughter’s classmate, Megan Meier, into committing suicide by bullying her (Megan Meier) [53]. In 2008, false rumors on the Internet in South Korea drove a famous actress, Jinsil Choi, to commit suicide. These are only two of the episodes that received wide media coverage.

2.9.5. DEFAMATION AND INVASION OF PRIVACY

The Internet can greatly amplify opportunities for defamation and invasion of privacy for individuals and organizations. There are several reasons for this. First is the global reach and speed of information dissemination. Second are the Internet-based business models. Many major Internet-based companies collect data from people, and use the data to target online ads. Internet-based companies include e-commerce vendors, search engines, portals, social Web sites, etc. [54-55].

The data include not only the standard demographic and lifestyle data, but also even Web surfing click patterns. Third, the collected data is subject to theft by the hackers and unscrupulous insiders. Many who join social networking Web sites seem willing to share all kinds of minutiae about their daily lives. However, many do not want people who are not their online friends to see their personal profiles and posts. Further, many people use social networking Web sites to lure children for sex. Major social networking sites, such as Face book and MySpace, have been working to provide better privacy controls on the personal profiles of their members [56].

Google includes the Street View feature in the Google Maps service. Street View shows the images of streets, along with the houses, buildings, cars, and people on the streets. Some of the minor side benefits include the apprehension of two muggers in the Netherlands, and the locating of a missing 9-year-old girl in the state of Massachusetts (the United States). However, Street View has stirred privacy concerns worldwide [57]. In some countries, such as the United States and China, naked people or people in compromising situations were shown.

In some countries, including South Korea, the Street View cars ended up collecting data from the wireless networks. Switzerland’s federal data protection and information commissioner banned Street View, on the grounds that it fails to sufficiently blur people’s faces and automobile license plates. Greece has refused to allow Google to collect Street View images until stronger privacy safeguards can be put in place. Some residents of the Buckingham shire village of Broughton, United Kingdom, blocked a Google Street View driver from entering the village to collect images.

Google also had to reshoot Street View images in 12 cities in Japan, so that images over the fences of the houses would not be included. Four Google Italy executives have been indicted in Italy on criminal charges of defamation and privacy violation [58]. As executives they were held responsible for a 3-min video posted to Google’s Italy site which showed four youths in Turin bullying a boy with Down syndrome.

2.9.6. AIDING CRIME

People post various ‘‘how to’’ techniques that may end up helping others to break the law. The techniques include how to make explosives, make drugs (narcotics), break into homes and buildings, hack, create and spread spam and malware, etc.

2.9.7. ILLEGAL ONLINE GAMBLING

Online gambling has become a US$29.3 billion business in 2010, of which Europe’s share is US$12.5 billion [59]. Online gambling sites include PartyGaming, Bwin, Sportingbet, BetonSports, 888, Virgin Gaming, Gaming Corp., BookMaker, etc. To evade prosecution and tax, many of the illegal online gambling businesses operate in places such as Costa Rica (BetonSports), Antigua (Sportingbet), Gibraltar (888, PartyGaming), etc. Using some of the money they made, they even lobbied with the United States government to legalize offshore online gambling. For a while, such major corporations as Merrill Lynch, Goldman Sachs, Morgan Stanley, Fidelity, etc. were investors in these businesses.

2.9.8 OTHER PERVASIVE REPREHENSIBLE BEHAVIORS

All of the things mentioned above, unless done by mistake or by not knowing or thinking about the consequences, are reprehensible. If the damage is serious enough, they are subject to law enforcement and litigation. There are many types of reprehensible online behaviors that people cannot appeal to law enforcement or litigate, and just have to live with. The reasons are, from the perspective of the law, there are too many people to prosecute, there are too many higher-priority offline crimes for law enforcement to deal with, etc.; and from the perspective of the people subject to the reprehensible behaviors, the case may not be strong enough in the eyes of the law, it may take too long to prosecute, it may be too costly to litigate, etc.

Many people buy items on Internet auctions, and do not pay, after taking delivery of the items; or many people get paid for items, but do not ship the items. Many people post comments to online discussion forums and bulletin boards in abusive language, and resort to personal insults and attacks. Many people post copyrighted materials without authorization, and post pornographic materials. Many people post to popular Internet sites truly depraved materials, such as graphic photographs of gang killings, animal abuse and twisted forms of pornography.

As a result, large Internet sites, such as Face book, MySpace, MSN, Yahoo, YouTube, etc. have to hire internal content reviewers or outsource content reviewing. They try to minimize truly depraved materials from staying on the site long. There are so many such materials that the content screeners need psychology counseling.

2.9.9 CAUSES OF THE MALWARE IN INTERNET

The nature of the Internet facilitates the dark side. First is the anonymity. Most Web sites require no more than a valid e-mail address for a person to become members. People can use a number of temporary e-mail accounts; pseudonyms in chat rooms, instant messages, text messages; and other means to hide their identities on the Internet [50]. Anonymity brings out the worst in many people: with no sense of guilt or inhibition, they launch personal attacks on others, post unfounded stories about others, spread false information, post or disseminate without authorization copyrighted digital properties, etc.

Second is the safety in numbers. For example, when millions of people post digital music, photos, or video without authorization on MySpace, Flicker, or YouTube, respectively, it becomes impossible for the authorities to enforce the copyright law. So, the sense of safety in numbers encourages people to scoff at the copyright law online.

Third is the availability of highly valuable but free contents and services, such as search engines (e.g., Google, Yahoo, MSN, etc.), free content services (e.g., many online newspapers, blogs, etc.), free software and storage services (e.g., Google Apps, Gmail, Hot Mail, etc.), social networking sites (e.g., Face book, MySpace, and Twitter), social media sharing sites (e.g., YouTube), even illegal file-sharing sites (e.g., Napster and Kazaa), etc.

Such free contents and services have led people to expect all contents and services on the Internet to be free of charge. This has led to the mass violation of the copyright laws online. There are at least four categories of reasons people join the dark side. They include desire for financial gains, psychology, changing mores, and miscellaneous. The desire for financial gains is the major reason for spam, click fraud, phishing, online frauds, violation of digital property rights, online pornography, online gambling, hacking, malware, etc.

Psychology has at least seven elements which are as follows, The dark side of people, Weak sense of reality, Need to reassure a sense of self-worth, Desire for self-promotion, One-upmanship, Desire to release stress and Sexual dysfunction,. Foremost among them is the innate dark side of people, including malice, and such negative emotions as anger, envy, jealousy, etc. The innate malice causes people to do bad things to others, knowing fully that they are harming real people.

The weak sense of reality causes a person doing bad things not be aware of the obvious fact that what he does or says affects real people, rather than inanimate Internet user accounts. This makes it easier for people, especially teenagers, to bully classmates, spread false rumors, spread malware, engage in click fraud, online frauds, identity theft, post comments in abusive language, post copyrighted digital properties without permission, etc.

The need to reassure a sense of self-worth leads some people to try hacking, spreading of malware, and denial of service attacks just to show off their technical prowess. It also causes people to spread false information to show off some special and intimate knowledge about something. The desire for self-promotion causes some people to spam to lure people to their blogs or portal cafes. One-upmanship causes some people to discover and disseminate ways to thwart measures for combating the dark side. This is one of the important reasons for the increasingly sophisticated techniques behind spam, malware, hacking, phishing, denial of service attacks, and click fraud.

Further, many people do bad things just to release stress and have fun (at others' expense). Sexual dysfunction is behind the sexual predators searching for preys online. With the passing of time, the mores of people change. What was unacceptable in a certain period becomes acceptable later. Examples include informal salutations to seniors, superiors, and strangers; casual attire in schools and work places; divorce; same-sex union; etc.

During the past decade, the globalization of the world, dissemination of new types of music (e.g., Gangsta rap), fantasy movies with computer-graphics-based special effects, movies with violence and sex (e.g., Friday the 13th, A Nightmare on Elm Street, Godfather, Terminator, Rambo), television shows, etc. have all contributed to the changing of the mores of people, especially the young people.

The sense of what is right, ethical, proper, respectful, or even real, has undergone changes. These changes are reflected in how people behave on the Internet. There are miscellaneous other reasons that contribute to the dark side of the Internet. Some people do so under the influence of alcohol or drugs, etc. Governments have political, military and law enforcement reasons to resort to hacking, phishing, denial of service attacks, and malware.

2.10 MALWARE RISK ASSESSMENT

Ubiquitous malware poses a threat to all computer users. The seriousness of the threat varies greatly. For some users, it is sufficient to install operating system updates regularly and utilize caution in running software found on the web. For organizations that face a high risk of being specifically targeted by a malware author, it is advisable to find computer security experts to defend their computers — or better yet, to simply avoid using networked computers for their most sensitive activities.

In early 2007, it was becoming readily apparent that a paradigm shift was taking place in terms of new malware releases. Instead of a small number of widely targeted, interspersed malware instances, there was a pronounced move towards a large volume of narrowly targeted, highly-related malware[25]. Accordingly, this influx of short-lived malware served to circumvent signature-based detection techniques through its sheer volume.

It was therefore not surprising to see malware detection becoming increasingly reliant on heuristics and other non-signature based techniques [25]. As such, the use of a sample-based approach for mapping vendor names to unique CME identifiers was no longer sensible for heuristically-identified malware instances. Consequently, the last CME ID issued was assigned to the malware instance commonly referred to as the “Storm” worm, and the CME-related efforts were transitioned to supporting the DHS/DoD/NIST Software Assurance Malware Working Group.

Based on the changing malware landscape and our experiences with CME, it became clear that a formal language was needed for communicating the fundamental characteristics of malware. Describing malware in such a fashion would mitigate the challenges posed by armoring and obfuscation techniques, which hinder the development of physical signature and heuristics. Likewise, it would allow for improved malware detection and mitigation by way of providing a common format for relating any information garnered through analysis or observation.

Dedicated Malware Solutions are Essential in Supporting UTM Appliances CALGARY, AB. and New York, NY. November 16, 2009 – Today at Interop NYC Wedge Networks (Wedge), the leader in high performance network based Web security solutions, is pleased to announce The Tolly Group has released a report that commends Wedge's BeSecure NDP-2040 Web Security Appliance and its unbeatable protection against malware attacks. The report also confirms that Unified Threat Management (UTM) devices cannot provide blanket protection and recommends a supplementary solution, such as Wedge Networks BeSecure, for maximum malware detection and blocking.

"The BeSecure Web Security Appliance performed extremely well in Tolly's real-world testing, maintaining high throughput with comprehensive malware detection and blocking," says Kevin Tolly. "When compared to a leading UTM solution, the results clearly show BeSecure is superior for malware detection and blocking."

Wedge Networks BeSecure NDP-2040 Web Security Appliance blocks 100 percent of Wild List viruses, 98 percent of VX (Virus) Heavens virus while Fortinet, the leading UTM vendor only block 86 percent of WildList viruses and 23.3 percent of the VX Heavens viruses. Wedge consistently outperforms the competitor in performance tests across a range of 10KB to 10MB payloads of clean and infected traffic. This large gap in performance and accuracy leaves organizations with only UTM appliances vulnerable to malicious attacks and virus outbreaks which can result in lost revenue, data and productivity.

Both The Tolly Group and Wedge Networks agree high scanning performance is ultimately based on high security accuracy. While today's UTM products provide effective fire walling capability, they have low malware detection rates with poor performance. By installing a complementary solution, such as BeSecure Web Security Appliance, companies will experience comprehensive anti-virus detection without compromising network performance.

"While it's essential for enterprises and service providers to deploy effective anti-malware solutions that protect the entire network, Next-Generation Firewalls (NGF) and UTM appliances do not provide complete security coverage for Web 2.0 threats," says Dr. Hongwen Zhang, president and CEO of Wedge Networks. "Organizations using these devices experience malware problems and don't understand why. Adding the BeSecure appliance to a company's IT infrastructure will offer the most accurate and manageable solution to protect a network." For more detailed results from The Tolly Group report and how a dedicate malware solution can increase the security of UTM appliances.

2.10.1 ABOUT WEDGE NETWORKS INC.

Wedge Networks provides high performance network-based web security solutions to organizations worldwide. Its technology leadership in deep content inspection allows companies to protect against new and emerging web-based threats that traditional scanning methods have difficulty intercepting and controlling. Specialized in malicious-code detection and filtering at the application layer, Wedge Networks' solutions touch millions of users and provide protection for utilities, service providers, health care, oil and gas, government, web sites, and many other customers from SMBs to enterprises. Wedge Networks is headquartered in Calgary, Canada and has international offices in San Jose, USA and Beijing, China.

2.10.2 ABOUT WEDGE BESECURE

Wedge Networks' award-winning BeSecure Web Security appliance allows enterprises and service providers the ability to detect, block and defend against web-based malicious attacks that steal data, disrupt networks and damage credibility. The BeSecure suite of products include: BeSecure Web Anti-Virus Module, BeSecure Anti-Spam Module, and BeSecure Webfilter Module.

2.10.3 ABOUT THE TOLLY GROUP

In over 20 years, The Tolly Group has published hundreds of independent tests, technology articles, and opinion pieces in leading business and trade publications. This high profile and unparalleled publishing record explains why The Tolly Group is viewed as an unimpeachable source by editors at Business Week, Computer world, Information Week, Network World and VAR Business among others.

2.11 PC TOOLS' NEW ADVANCED RESEARCH TEAM IN BOULDER

PC Tools' new advanced research team in Boulder, Colorado - which focuses on behavioral technologies - have identified a number of key trends that render signature detection ineffective in combating current and future malware threats. "The security space is changing rapidly. We are witnessing a major shift in the anti-malware market place moving into a new era of Malware 2.0.," said Kurt Baumgartner, chief threat officer. "We are now dealing with zero-minute, rather than just zero-day exploits that have the potential to further evade signature detections," said Baumgartner.

Three key trends identified include:

* Malware variants are now released at immense rates, driving up sample volumes and making it almost impossible for researchers to keep on top of updates using manual analysis. These threats are taking advantage of the non-detection sweet spot where they can freely propagate and infect before anti-malware companies can respond.

- New compilers and other techniques are being used to make threats more difficult, if not impossible, to detect with traditional signature-based systems. This technique relies on advanced server-side systems to create completely unique threats each time, devoid of the commonalities required for signature detection to be effective.

* ‘Micro-malware’ - thousands of malware variants - are in circulation, but are focusing attacks on smaller groups of PCs, making it less likely to attract the attention of security vendors. As a result, malware is spreading in epic proportions and security vendors are being forced to triage the samples.

“These three key trends demonstrate that, just as the internet has moved into the Web 2.0 phase, the security space is moving into a new era of Malware 2.0. The real challenge for security vendors is in identifying new ways to detect the behavior of malware. Signature identification alone is ineffective in protecting consumers,” said Baumgartner.

With the spyware industry estimated to be worth billions of dollars there are significant incentives for malware authors to develop techniques to avoid detection.

“We estimate that one-in-five users with major anti-virus products already installed on their computers are still vulnerable to these new and emerging threats,” Baumgartner said.

“The results of internal testing on the most commonly used security software found that the addition of behavioral detection increased the effectiveness of traditional antivirus technology by up to 126 percent. In every case, each of the popular products tested missed a large quantity of in-the-wild threats active on users’ PCs,” Baumgartner said.

System for combating malware with keystroke logging functionality: A method is carried out by a computer system for combating malicious keystroke-logging activities thereon. An operation is performed for generating a plurality of fake keystroke data sets that are each configured to resemble a keystroke data-set generated by keystrokes made on an input device of the computer system while entering sensitive information of a prescribed configuration. An operation is performed for receiving an instance of the sensitive information instance of the prescribed configuration concurrently with generating the fake keystroke data sets.

Receiving the sensitive information instance includes a user of the computer system entering the sensitive information instance by performing keystrokes on the input device of the computer system such that a real keystroke data set corresponding to the sensitive information instance is generated. An operation is performed for embedding the real keystroke data set within at least a portion of the fake keystroke data sets after receiving the sensitive information instance. Kaspersky Lab has patented a groundbreaking piece of technology in the USA that allows the potential scale of malware epidemics to be accurately predicted in order to prevent them from spreading.

Today's malware has the capacity to spread like wildfire, with millions of computers infected in an instant as an epidemic sweeps across the Internet. This can take down huge swathes of infrastructure, bringing information highways to a standstill and leaving systems vulnerable to data leakage which in turn opens the door to large scale fraud. Detecting malware on every computer that is infected during an epidemic has little or no effect.

What is needed is a reliable method for estimating the potential scale and direction of an epidemic, an early warning system, and that is exactly what the new technology developed by Kaspersky Lab's Yury Mashevsky, Yury Namestnikov, Nikolay Denishchenko and Pavel Zelensky, is capable of doing.

The technology was granted Patent No. 7743419 by the US Patent and Trademark Office on 22 June, 2010.

The patented new technology works by analyzing statistical data about threats received from a global monitoring network. The network tracks malware downloads hacker attacks and other similar security incidents, recording the times that they occur, their source and geographical location etc. Emerging epidemics can then be identified by the number of incidents occurring during a specific period in one location or another. This method makes it easy to pinpoint the source of an epidemic and forecast its likely propagation pattern.

Protective measures can then be developed and implemented by those countries in the path of the epidemic, slowing the proliferation rate considerably and providing effective damage limitation. The monitoring, detection and analysis of data is performed in real time, making the patented technology especially effective against malware epidemics that spread rapidly.

"The new system has a number of advantages over other similar solutions. This technology contains a subsystem for tracing the source of the threat, a module that generates protective measures and a subsystem that simulates the spread of an epidemic," noted Nadia Kashchenko, Chief Intellectual Property Counsel at Kaspersky Lab.

Kaspersky Lab currently has more than 50 patent applications pending in the USA, Russia, China and Europe. These relate to a range of unique information security technologies developed by the Company's personnel.

2.12 DHS/DOD/NIST MALWARE WORKING GROUP

Shortly after the issuance of the last CME ID in early 2007, the DHS/DoD/NIST Software Assurance Malware Working Group was stood up with representatives from MITRE and the Anti-Spy-ware Coalition (ASC) as co-chairs. The Malware Working Group was created to develop a framework of descriptive attributes of malware in order to:

- Improve communication by characterizing patterns (attributes and behaviors) to identify and describe malware types and instances
- Enable users to make informed decisions, i.e., lead to more stringent user acceptance criteria if potentially malicious code is to be installed with user knowledge
- Enable legal definitions of malware, spyware, adware
- Provide objective criteria for anti-malware tool assessments

2.13 DESCRIPTIVE AND DETECTING MALWARE

This work by the Software Assurance Working Group laid the foundation for the ‘Descriptive And Detecting Malware’ effort, with the initial focus being on a single use case, namely attempting to develop a legally defensible definition of malware.

This was driven by the desire to control the installation of spyware and adware and the ASC‟s success in using the concept of "potentially unwanted behaviors" to put the discussion in a more neutral setting. In talking to members of the security community involved with malware, we saw the need to broaden the Malware Working Group's set of use cases in order to provide greater uniformity in reporting malware, standardize the results of malware analysis, and support the development of databases describing adversary tactics, techniques, and procedures.

CHAPTER 3

TECHNOLOGY-CENTRIC DARK SIDE OF INTERNET AND OBJECTIVES

TECHNOLOGY-CENTRIC DARK SIDE OF INTERNET

Technology is responsible for the rise of the dark side of the Internet. Some of the elements of the dark side have no offline counterparts. Technology has been used to fight them. Unfortunately, technology has been caught in a perpetual catch-up cycle against the resourceful and creative cyber crooks. Each element of the technology centric dark side has one or more fundamental difficulties that make perfect technology-centric solution impossible. For example, there are millions of computers that are infected with malware. Hackers use these computers to launch spam, malware, denial of service attacks, phishing, and click fraud.

The presence of such computers also makes it very difficult to prevent or trace the true sources of spam, malware, and denial of service attacks. It is very difficult to filter spam that use images. Further, there are many sources of technologies on the Internet that aid cyber crooks. These include software that enable spammers to amass e-mail addresses, defeat anti-spam technologies and launch spam on a massive scale. It is difficult for people to detect a phishing attempt by just the look and feel of the fake Web sites.

Non-technology responses are required to address each element of the dark side, regardless of whether it is technology-centric or not. The targets of technology-centric dark side include individuals, commerce sites, institutional sites, government sites, and nation’s operational infrastructure sites. The individual targets include personal home pages, e-mail accounts, social networking site accounts, etc. Commerce sites include all corporations with Internet presence.

Institutional sites include financial institutions, hospitals, schools, transportation companies, etc. Government sites include sites for all branches, police, intelligence agencies, and the military. Infrastructure sites form a nation’s operational infrastructure, such as the power grid, financial systems, transportation systems, hospital systems, etc. They overlap with some institutional sites and government sites.

In this, we discuss each element of the technology- centric dark side, how it works, and the technology responses to it.

3.1. SPAM

The origin of the term spam is not clear. Possibilities include “stupid, pointless and annoying messages”, “self-promotional messages”, “solicited pornography and marketing”, “shit posing as mail”, etc. There are several types of spam: e-mail spam, social spam, messaging spam, Internet phone spam, etc. Social spam includes search engine spam, social networking spam, and social Web spam. Social Web spam in turn includes spam on bulletin boards, Internet forums, news groups, blogs, wikis, etc. As the messaging spam and Internet phone spam are not pervasive, we do not consider them in this paper.

3.1.1. E-MAIL SPAM

Gary Thuerk sent a Digital Equipment Corp. product presentation announcement on May 3, 1978 to all 600 users of the ARPANET (now the Internet), reaching half of the users. That was the first spam e-mail in history. No user of the Internet can escape spam e-mail. According to some estimates, there are 200 billion spam e-mails a day; 85% of all e-mails is spam; 9% of spam e-mails is scam e-mails; and 81% of the spam e-mails are “pharmacy ads” [61-62] The cost of sending spam e-mails is considerably less than that of regular mail. Most of the cost of sending spam e-mails is borne by the recipients.

To add insult to injury, because of the spoofing of senders’ IP addresses, the recipients may even be black-listed as spammers. About 100,000 individuals and businesses are the largest senders of spam e-mails. There are about 200 master spammers who are responsible for a large proportion of the spam e-mails.

Each master spammer sends 10–100 million spam e-mails a day. A spammer makes an upfront investment of about $10,000 on (up to 15) computers, software (to harvest e-mail addresses, and hide his identity), and lists of e-mail addresses to spam ($1200 for 10 million addresses). The response rate for spam e-mails is less than 0.1%, and so spammers have to work hard. They make $1000–10,000 a week: $60 for $150 Viagra order, $12 for mortgage lead, $5 for insurance lead, etc.[63,64].

3.1.1.1 HOW IT WORKS

Spammers are not necessarily technology experts. They receive lots of help from the spam supporting industry. The spam supporting industry includes spamware companies, list merchants, spam ISPs (Internet service providers), spammers’ convention, and spammers’ social networks. Spamware companies create, collect, and distribute spam-aiding software. Spam-aiding software include tools for harvesting e-mail addresses, tools for hiding spam sender’s identity; tools for defeating anti-spam technology; tools for launching spam e-mails; etc. Tools for harvesting e-mail addresses gather e-mail addresses from Internet chat rooms, Web sites, news groups, e-mail address books, public databases, etc.

Tools for hiding spam sender’s identity generate random e-mail addresses for the FROM: and REPLY TO: lines in the spam e-mails. Tools for defeating anti-spam technology insert fraudulent headers or spam-filter-defeating letters or images into e-mail messages. Tools for launching spam e-mails use many mail servers simultaneously, and use open relays and zombie computers. (Zombie computers are computers that are infected with backdoor worms.

A backdoor is a program or means that enables bypassing normal authentication; it may also be a modification to an existing program or hardware device [65].)

List merchants harvest e-mail addresses and sell them. Spam ISPs provide services that enable a variety of cyber crimes, such as spam, botnets, pornography, and malware distribution. Many spam ISPs have been identified and shutdown, including InterCage, Atrivo, McColo and 3FN (Pricewert). When McColo was shutdown, there was said to have been a 50% drop in spam e-mails, albeit for a very brief period. Some of the shutdown spam ISPs may have moved their command and control servers to other countries, and new spam ISPs have sprung up [66].

Spammers have used open relay servers to transmit spam e-mails. The SMTP (Simple Mail Transfer Protocol) is the standard in wide use today for e-mail transmission across IP (Internet Protocol) networks. It is designed to use relay servers to deliver messages from one host to another. An open relay server accepts messages no matter which host it comes from. Spammers also use networks of zombie computers (i.e., botnets) to launch spam e-mails. Spammers may even be funding the creation of worms. Some worm creators have been caught selling lists of IP addresses of zombies [67].

3.1.1.2. ANTI-SPAM TECHNIQUES

Many anti-spam, or spam filtering, techniques have been developed. The results of testing representative anti-spam products by Opus one [68] in February 2007 showed that the average spam filtration rate was about 86.93% and the average false positive rate was 0.3% [69]. Various vendors offer total malware solutions that include anti-spam techniques. These include McAfee Internet Security 2010 and McAfee Total Protection 2010 [70], Kaspersky Internet Security 2011 [71], etc. Fig. illustrates the spam-filtering process.

The tokenized splits incoming e-mails into tokens. Then the analysis engine filters e-mail when the e-mail address of the sender is on the blacklist or a spam feature is found in the title or content of the e-mail. Spam features include emphasis signs, such as capital letters and exclamation points; money value that appears in ads, expression of discount or sale; repetition of special symbols with no meaning, and non-regular words (e.g., cl!ck, 0ther, v/agra). (We will discuss spam-filtering techniques further later.)

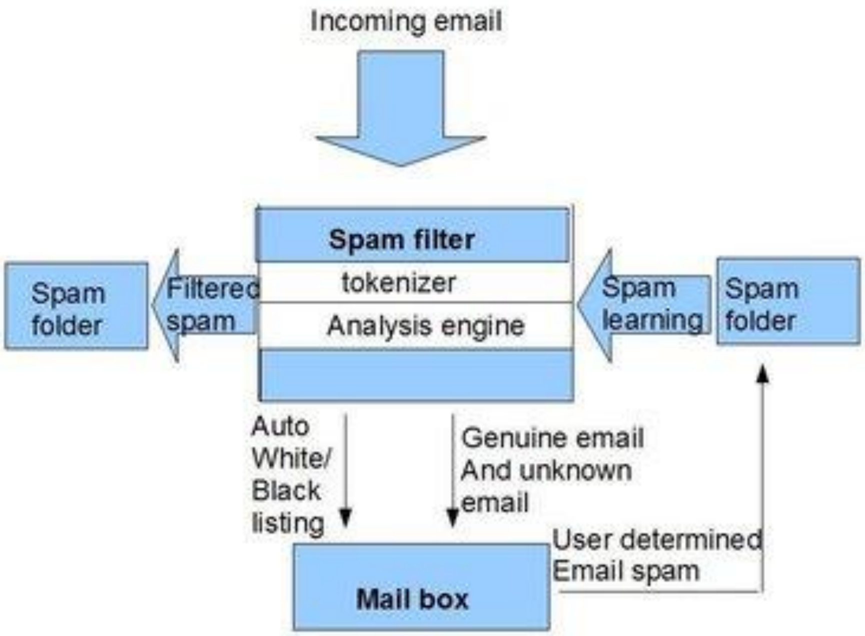


Fig.1. E-mail spam-filtering process.

The addresses of e-mails that contain spam features are automatically added to the blacklist. The user can move unfiltered spam e-mails to the spam folder. The analysis engine continually learns from spam features from such e-mails. Anti-spam techniques may be classified into at least seven categories, as follows:

Directory based filtering: This technique uses a blacklist and white list of e-mail addresses to have e-mail server's block or pass e-mails. The problem with this technique is that the lists must be continually updated, as spammers continue to change their e-mail addresses.

Rule-based filtering: This technique analyzes the sender address, return address, subject header, and message in e-mails for known spam words. Spammers have continually come up with ways to get around this technique. They include "undelivered mail notification" in the subject header; auto-generated sender names in the From: field; and auto-generated return address in the Reply To: field.

For example, to get around the spam filtered word "Viagra", they misspell the word or insert other characters (e.g., Vlagra, Via'gra, Vi@graa, etc.). They insert comments in between letters in HTML. Content-based filtering: This technique attempts to computationally distinguish between a spam e-mail and a legitimate e-mail using machine learning techniques. The machine learning techniques decision tree, naı¨ve Bayesian [72], SVM (Support Vector Machine) [73], and artificial neural networks.

Spammers defeat this technique by inserting irrelevant words in e-mail messages. They also defeat this and any technique based on text analysis by inserting GIF or JPEG or animated GIF images in the messages. They also contort the shapes of the letters in the image (as in CAPTCHA). They also write text in white font on a white background to evade human detection. Image recognition technology is not mature at this time, and image-based spam poses a serious challenge.

Collaborative filtering: This technique categorizes e-mails received by, say, user B as spam if the e-mails were reported as spam by, say, user A on the same e-mail server. The problem with this technique is that it can only be used for the same e-mail server, since different e-mail servers do not share information. Further, spammers may generate many versions of the same spam e-mail by including random words, so that they are recognized as different e-mails. They may also continually change their e-mail or server addresses.

Spam classification through social network: This method is based on social network analysis [74]. It represents e-mail senders as nodes on a graph, and the relationship between people who send and receive e-mails as a link on the graph. Nodes that do not belong in the linked list on the graph are regarded as spammers. The problem with this technique is that it requires tracking down e-mailing history, which can only be done in the same e-mail server; and it classifies as spam all e-mails from first-time senders.

Domain authentication: This technique is used to verify that an e-mail sender's domain has not been fabricated or spoofed. It is easy for a sender to masquerade as someone else because of the openness of the SMTP protocol. Spammers often take advantage of this weakness to maintain anonymity. AOL's SPF (Sender Policy Framework), Microsoft's Sender ID, Yahoo's Domain Keys, and Cisco's IIM (Identified Internet Mail) are protocols to authenticate the sender's address. SPF (Sender Policy Framework) has evolved from these protocols, and is widely used. It uses a reverse lookup method on the MX record of the DNS to verify the sender's domain. The receiving mail server extracts the domain address of the sender and sends a query to the DNS. The reply from the DNS contains the sender IP address which has been authorized by the sending e-mail server.

Link structure analysis: It is difficult to analyze the content of a spam e-mail if it contains only a small amount of text. Therefore, this method distinguishes between authentic e-mail and spam e-mail by first calculating the number of links that a Web document has been linked by other Web documents and the number of hyperlinks contained in the e-mail as well as the documents that those hyperlinks lead to. It then learns by using a decision tree. It also uses an altered link structure analysis algorithm using hyperlink server addresses.

There are at least five problems that fundamentally limit anti-spam techniques:

Anti-spam techniques are based on text understanding and image recognition technologies. Since spam e-mails contain messages in text and/or images, text understanding and image recognition technologies cannot be avoided. However, these technologies have fundamental limits that most likely will not be overcome in the foreseeable future. Today's e-mail systems are based on the SMTP, which was designed with absolutely no consideration of today's huge e-mail traffic, cyber crooks and their exploitations.

There are many open relays, proxy servers, and botnets; and it is virtually impossible to eliminate them. A great majority of computer owners are not even aware that their computers have been infected; they do not apply anti-malware techniques diligently; and there are malware that anti-malware techniques cannot even detect. The spam-aiding industry has successfully waged a oneupmanship war against anti-spam technology products.

Governments have so far been unable to eliminate it. Anti-spam products have to meet the dual requirement of penalizing spam and allowing legitimate marketing and freedom of speech. Some people, more than just the 0.01% of the recipients who become customers, find spam useful. The line between legitimate marketing campaign and spam is not clear-cut at some point in the spectrum of e-mails. This difficulty is similar to that of drawing the line between pornography and art.

3.1.2. SOCIAL SPAM

Search engine spam aims to fool search engines into assigning a high rank in the search results to a spam Web page. The spammer makes this happen by, for example, including popular keywords (e.g., iPhone) as tags for his Web page (e.g., a blog or ad on, say, flower arrangement). As a result, users looking for Web pages about iPhone end up with a link to the blog or ad on flower arrangement. Social networking spam is associated with social networking services, such as Face book, MySpace, Twitter, LinkedIn, etc. Members spam other members. Some spam to generate new visitors to their personal blogs or home pages. Some spam to market something. Some spam for other reasons.

3.1.2.1.HOW IT WORKS.

There are two types of techniques for generating search engine spam: content-based and link based [75]. Content-based spam includes popular keywords not related to the spam Web page. Link-based spam inflates the number of pages linked to the spam Web page. It exploits the HITS (Hyperlink-Induced Topic Search) algorithm, which determines search result rank by calculating the authority and hub scores of a Web page. Intuitively, a good hub is a page that points to many other pages, and a good authority is a page that is linked by many different hubs. A search engine that uses the HITS algorithm to rank pages returns as query result a combination of pages with the highest hub and authority scores. The spammer would add many outgoing links to the target Web page to inflate its hub score. He can spam the authority scores by adding incoming links from important hubs to the target Web page [75].

Techniques used to create content-based spam include keyword stuffing, meta-tag stuffing, and article spinning [76]. The keyword stuffing technique inserts keywords into a Web page. This produces Web pages that can easily appear at the top of the search results. The meta-tag stuffing technique repeats keywords in meta-tags and inserts new meta-tag keywords into Web pages. The article-spinning technique works by rewriting existing Web pages.

Search engines, when calculating search result ranking, assess penalty for simple repetition of keywords. The article-spinning technique avoids such penalty. The link-based techniques insert links into a Web page. They include link building software, link farms, hidden links, spam blogs, page hijacking, etc. [76]. On social networking sites, spammers create false personal profiles, collect information on other members, create malicious links, and phish.

Spammers send messages and write replies to other members with embedded links to Web pages of their choosing. They manipulate survey results and/or disseminate unreliable survey results for a wide range of topics, such as popularity of products, people, organizations, etc. They do this by manipulating click-through rates by, for example, continually voting with new IDs and/or inviting friends to do likewise, etc.

3.1.2.2. ANTI-SOCIAL-SPAM TECHNIQUES.

There are two techniques to dealing with search engine spam. One is to increase the accuracy and reliability of search results by neutralizing the interference from search engine spam. This takes into consideration the different types of search engine spam. Example approaches include Trust Rank [77] and Spam Rank [78]. Another approach is to detect spam for each spam generation technique and spam environment. For example, links between Web pages are analyzed to detect link-based spam [79], while keyword analysis and comparison with legitimate documents are done to detect content-based spam [80].

In addition, there are spam detection techniques specifically designed for spam that occur in particular environments, such as blogs [81]. HubSpot Small and HubSpot Medium by HubSpot Ltd. help users use and measure spam detection techniques [82].

There are several techniques for dealing with social networking spam. First is to check the spammers' IP addresses. Second is to block spammers who use specific spam words. Third is the use of a reverse Turing test to determine if a user used a spam tool or personally wrote comments. Fourth is to disallow links in posts. When spam terms are inserted in a user's blog or personal Web site, an immediate notification can be sent, the relevant terms deleted, and the person responsible is recognized as a spammer, and appropriate actions taken.

Technology responses to social spam boil down to understanding the meaning of the text in Web pages. Search engine spam includes keywords or links that are not related to the Web pages. Social networking spam includes inaccurate or fabricated posts, and the relationships formed online are sometimes based on falsified or inaccurate personal profiles of the members. Because of the limitations of text understanding technology, it is very difficult to determine whether the injected keywords or links are related to the Web pages. Further, for the same reason, it is also very difficult, if not impossible, to determine the veracity of the personal profiles and personal posts in social networking Web sites.

3.2. MALWARE

Malware includes viruses, worms, Trojan horses, spyware and adware. The payload of malware (i.e., what malware delivers or does) has apparently been largely benign. However, some malware creators have started creating highly sophisticated malware and planted them in millions of computers globally. Conficker, apparently disseminated from November 2008, is now said to control 7 million computers in more than 200 countries [83].

It includes defense mechanisms that make it hard to erase it; it hides from or even kills programs designed to look for it.

The Kneber botnet, using the ZeuS Trojan, can nowaccess 68,000 corporate computers; more than 2000 digital security certificates; many e-mail systems, online bank accounts, and social networking Web site accounts [84]. One volunteer security organization is currently monitoring 5,900 separate botnets of zombie computers [85].

Malware are becoming increasingly complex, and their ability to hide has become so sophisticated that even professionals have a difficult time understanding the infected state of a system. Further, there is a diversity of methods for spreading malware, which include e-mail, exploitation of software flaws, and USB. It has become difficult to figure out how the infection occurred.

3.2.1. HOW THEY WORK

Malware can be installed in a computer system in a number of ways. It may come on a USB drive that seems to have been accidentally lost [85]; it may enter as part of software downloaded from a file-sharing network; it may enter in the form of an ActiveX control on a Web site; it may come in an e-mail attachment; it may be installed by exploiting flaws in a Web browser, media player, messaging client, etc. [86].

3.2.1.1. VIRUSES.

A virus is code that attaches itself to a host (e.g., a program file, or a boot record on a hard disk) in a host computer. When a user runs the infected host, he causes the virus code to run also. A virus can spread to a different computer when the infected files are taken to the computer, for example, as an e-mail attachment or on removable storage, such as USB, CD, and DVD. The effect of a virus ranges from simple annoyance to damages to the hardware, software, or files. Viruses may be classified as resident or non-resident based on how they infect files [87].

A non-resident virus searches for new files to infect and infects those files. A resident virus, upon execution, loads into memory a module that infects files, and ensures this module to be executed (and infect files) each time the operating system performs a certain operation [88]. Some resident viruses are designed to infect all possible hosts with which they come in contact. For example, such a virus can attach itself to a virus scan program which checks all files in the computer system, and infect all files that are being checked.

Other viruses are designed to infect hosts sporadically, for example, only when files are being copied. Viruses can infect various hosts in a computer system, such as binary executable files, volume boot records of disk partitions, master boot record of hard disks, general-purpose script files, application-specific script files, system specific autorun script files, documents including macro, and cross-site scripting vulnerabilities in Web applications [89].provides a good discussion of the one-upmanship game that goes on between virus creators and anti-virus software developers. Many viruses are designed to avoid detection by anti-virus software.

Below we summarize the techniques used.

1. Some place the virus code on unused areas of the files to infect, thereby not increasing the size of the files.
2. Some kill tasks associated with anti-virus software.

3. Some avoid infecting anti-virus software, since many anti-virus software perform an integrity check of their own code and can detect attached viruses.
4. Some avoid infecting bait files, which are created by anti-virus professionals to capture and analyze viruses.
5. Some intercept file-read requests from anti-virus software to the operating system, and return the uninfected version of the files to the anti-virus software.
6. Some modify the virus code on each infection, so as not to leave a single virus signature for anti-virus software to detect.
7. Some encrypt the virus code to make it difficult for anti-virus software to determine the virus signature.

Microsoft software, due to their market dominating position and many defects and weaknesses, have been the main target of attack. Microsoft's networking software (such as Microsoft Outlook and Internet Explorer), integrated and individualized application programs (such as Microsoft Office), and general-purpose application programs written in script language (such as Visual Basic Script) are especially vulnerable to attacks [87]. All operating systems that allow third-party programs to run can run viruses. Some operating systems are comparatively more secure than other operating systems. For example, operating systems based on Unix (or NTFS recognition program of Windows NT based platform) allow user programs to run only on protected memory spaces.

3.2.1.2. WORMS.

Unlike viruses, worms can replicate themselves and send their copies to other computers on the networks, without human aid. Worms spread by taking advantage of file transport features on the communication system, exploiting vulnerabilities in operating systems [88]. A worm may replicate itself and send out hundreds or thousands of copies of itself. For example, a worm may send a copy of itself to everyone listed in a user's e-mail address book. Then, the worm replicates and sends itself to everyone listed in each of the receiver's address book, and so on [88].

Because of the ability to replicate itself and send the copies across networks, the worm consumes system memory and network bandwidth, thus overloading Web servers, network servers, and individual computers. For example, worms like Morris and Mydoom can drastically slow down the Internet, if scattered to the maximum.

The payload of a worm may be code to delete files on a host system (e.g., the ExploreZip worm), encrypt files, or send documents via e-mail. It can also install a backdoor in the infected computer, turning the computer into a zombie. Examples of worms that create backdoors include Sobig, Mydoom and Blaster Worm [89]. Back doors can be used by other worms. For example, Doomjice uses back doors created by Mydoom to spread itself [89].

3.2.1.3. TROJANS.

Unlike viruses and worms, Trojans neither reproduce by infecting other files nor self-replicate. The Trojans appear to be useful software, but, once installed or run on a host computer, inflict damage. The damage ranges from files and passwords being logged and transmitted, to files and information being deleted from the host computer.

Further, it may result in the installation of a backdoor, turning the host computer into a zombie to be controlled by hackers [65]. Recently, a type of Trojan horse called Trojan Downloader has appeared, which downloads more Trojans from certain Web sites. This kind of program
is particularly dangerous, since unknown additional programs can break into the computer.

3.2.1.4. SPYWARE.

Spyware was originally created by an Internet Ad Company named Radiate Group, but later came to be changed and misused. Spyware are often designed to capture user’s key strokes, or copy data from hard drives, and transmit the data. In this way, spyware steal sensitive information, such as user ID and password, log contained in the cookie, data in user’s favorite Web sites, etc.

The effect of some spyware is annoying; for example, they fix Internet Explorer’s home page, and ignores the user’s attempts to unfix it. In general, however, the effect of spyware has proven to be costly and troublesome for the victims. Often the only way to deal with spyware has been to back up all user data and reinstall the operating system.

Spyware get installed on PCs, without their owners being aware, through software downloads, clicking on Web sites and e-mail attachments. Spyware often exploit flaws in the Internet Explorer and ActiveX [90]. Fraudsters have also tricked people into downloading ‘‘anti-spyware’’ programs that either do not remove spyware, or actually install real spyware.

3.2.1.5. ADWARE.

Adware refers to malware that spreads by being tied with ads or contains ads within it. Adware are similar to spyware. However, adware is installed with the consent of the user, when the user clicks on the agreement for installing Active X control while accessing a particular Web site. The user gives his consent usually not knowing what the Active X control exactly does, and not even bothering to read the terms of installation.

Like spyware, adware may record data about users’ activities without their consent, and sometimes the data is re-sold. Adware installed may fix the home page to a certain Web site; display unsavory ads; and redirect to a certain Web site. Adware may also disable the Internet or cause system error. Income from ads is an incentive for programmers to create, maintain, and upgrade adware. Some adware are shareware.

3.2.2. ANTI-MALWARE PRODUCTS VIRUS SCAN PROGRAMS FIND VIRUSES IN ONE OF TWO WAYS

The first and most common virus detection method is to look up a list of virus signature definitions. It checks for viruses hiding in a computer’s memory (RAM, booting sector, etc.) and/or in fixed or removable drives (hard drives, floppy drives, etc.) by comparing to a database of signature definitions of known viruses.

The second method is to use a discovery algorithm to find viruses based on the computer's behavior. This checks files being moved, such as e-mails, and files that are open. One problem with the virus scan programs is that they require frequent updates to add new virus signature definitions or updated algorithms [91].

The speed with which variations of malware are being born is faster than the ability of security researchers to find cure for newly discovered malware [92]. Computers around the world are vulnerable to infection by thousands of newly created malware if update is delayed even for a single hour. As such, most companies currently offer virus signature updates at least once a day (as opposed to once a week before).

However, it is difficult to solve the current problems just by shortening update periods [93]. Many anti-virus product developers are using such methods as heuristic detection, proactive prevention, and sandbox in order to increase the rate of diagnosis. However, wide use of anti-virus products is hampered by the diversity of users' computer environments, computer settings, management of updates, and misdiagnosis. In addition, the size of anti-malware software increases as the number of diagnoses increases. This takes up memory space, which in turn slows down scanning speed.

There are many anti-virus products, for example, V3 [94], Norton AntiVirus and Norton 360 version 4.0 by Symantec [95], Kaspersky version 7.0.0.43 [71], Active Virus Shield by AOL [96], ZoneAlarm with KAV Antivirus [97], F-Secure 2007 [98], McAfee Active VirusScan and McAfee AntiVirus by McAfee [70], etc. Anti-worm products include VirIT explorer Lite developed by TG Soft [99], Comodo Internet Security developed by Comodo Solutions Ltd. [100], Brothersoft Community by Brothersoft.com [101], etc.

Anti-Trojan products include TrojanHunter by Mischel Internet Security [61], Trojan Remover by Simply Super Software [103], The Cleaner by Moosoft [104], etc. Anti-spyware products include Microsoft Windows Defender, Lavasoft's Ad-Aware SE [105], Patrick Kolla's Spybot—Search & Destroy [106], PC Tools's Spyware Doctor, Sunbelt Software's Counterspy, Trend Micro's HijackThis, Webroot Software's Spy Sweeper, ParetoLogic's Anti-Spyware and XoftSpy SE. Many anti-adware products contain antispyware capabilities.

Examples include Ad-Aware Free Internet Security by Lavasoft [105], Anti-Spyware Adware by Windows Registry Software [107], Brothersoft Community Toolbar by Brothersoft.com [101], L-Histidine by Shine-Star.com [108], McAfee AntiSpyware by McAfee [70], Norton Internet Security [54], Kaspersky Anti-Virus 2011 by Kaspersky Lab [71], etc.

Total anti-malware products include McAfee Total Protection 2010 and McAfee AntiVirus Plus 2010 [70], Norton Internet Security 2010 and Norton AntiVirus 2010 by Symantec [95], Kaspersky Anti-Virus 2011 by Kaspersky Lab [71], etc.

Computer security product vendors are starting to develop cloud-based anti-malware techniques. These involve a database of virus signature files managed by servers in the cloud (i.e., remote third-party managed data center).

The users' computers can query the database to determine whether a certain file is malicious. The servers in the cloud use the basic information for a file, digital signature analysis, reputation analysis, activity trend analysis, behavior-based activity analysis, relation between files and various other techniques to determine whether a file is safe [109].

The rate of diagnosis can be significantly improved when the servers are able to analyze over thousands of malware and collect as well as diagnose them in real-time. Further, with the cloud-based techniques, when a new file is created or a file is updated, the changes become available to the users' computers in real-time. This can substantially reduce the danger due to the delay between analysis of new malware and its availability to the users.

Fig. 2 shows how anti-malware products operate in the cloud. When the user accesses a suspicious file, he notifies the product in the cloud (data center). The product analyzes the file, creates a mitigating pattern file, and disseminates it among the servers within the cloud.

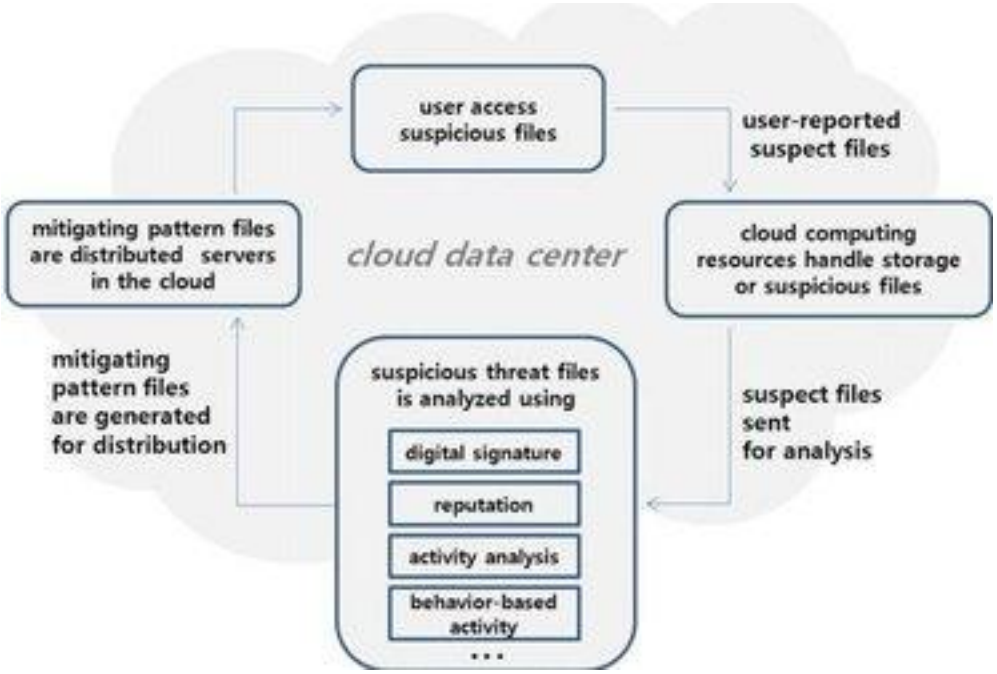

user access suspicious files
user-reported suspect files
mitigating pattern files are distributed servers in the cloud
cloud data center
cloud computing resources handle storage or suspicious files
mitigating pattern files are generated for distribution
suspicious threat files is analyzed using
digital signature
reputation
activity analysis
behavior-based activity
...
suspect files sent for analysis

Fig. 2. Anti-malware product in the cloud.

The explosive growth of malware poses major challenges to developers of anti-malware products. There are two fundamental problems with technology-centric products to malware. First is that the sheer number of malware variations and the speed with which they are being unleashed simply overwhelm the collective abilities of the security researchers and security product vendors to analyze malware and provide cures.

Malware creators are now creating and selling tools that can mass produce malware with the capability to mutate automatically [110]. Second is that a great majority of the owners of computers are not technically savvy enough or careful enough or diligent enough to prevent their computers from being infected with malware or remove the malware. Besides, most of the anti-malware products available today do not detect all malware.

3.3. HACKING

Hacking using the Internet dates back to the early 1980s [111]. A hacker group called 414s was arrested by the US FBI (Federal Bureau of Investigation) for breaking into computer systems of the Los Alamos National Laboratories and (New York City) Manhattan's Memorial Sloan-Kettering Cancer Center. In the late 1980s and 1990s, Kevin Mitnick broke into computer systems of major companies including Nokia, Fujitsu, Motorola and Sun Microsystems, causing approximately US$80 million worth of damage to these companies.

Computer systems of government agencies were frequent target of hacking. It was reported that there were 250,000 attempts to break into the Department of Defense computer systems in 1995 alone. In 2001, Gary McKinnon hacked into the US NASA (National Aeronautics and Space Administration) and US military computers, including ones operated by the Pentagon. In 2009, hackers, presumably in China, launched a massive attack on Usbased companies, including Google, Adobe Systems, Juniper Networks, Rackspace, Yahoo and Symantec. The purpose of this cyber attack was to gain access and modify source code repositories of these companies [112].

3.3.1. HOW HACKING IS DONE

Hackers often exploit weak authentication or flaws in popular software (e.g., the Windows operating system, SQL-based database systems, e-mail system) to break into computer systems. They also resort to such methods as password cracking, packet sniffing, phishing, malware, etc. to acquire passwords. In the following, we discuss only the exploitation of weak authentication and software flaws.

3.3.1.1. EXPLOITATION OF WEAK AUTHENTICATION.

In January 2009, an attacker gained the password of a Twitter staff member by password guessing. He succeeded, because Twitter allowed unlimited login attempts, unlike many systems that do not allow more than three attempts. Failure to provide sufficient authentication is one of the most common vulnerabilities in a system or Web site. Many systems allow short passwords, alphabet-only passwords, use of common words as passwords, etc.

Hackers and others often correctly guess the user names and passwords to break into computer systems. Many people invite hacking by using easy-to-guess passwords, such as the names of the user or the user's family members; or common words like "12345", "abc123", "iloveyou", "password", "qwerty", etc. [113]. Many systems also do not require the passwords to be changed periodically. Some Web sites allow the bypassing of authentication by modifying parameters inside the cookie, such as "loggedin= true". Systems with weak encryption of passwords are vulnerable to pre-computation attacks, where pre-computed hashes of a large number of words are compared against the hash of a user's password.

3.3.1.2. EXPLOITATION OF SOFTWARE FLAWS.

The following summarize known methods of hacking that exploits software flaws (vulnerabilities or security holes).

(1) Exploitation of memory safety violations:
Memory safety violations include buffer overflows and invalid pointer references [114]. These have been known for decades, but still they are pervasively used by hackers in compromising computer systems. The Morris worm [115] used buffer overflow on UNIX to propagate itself throughout the Internet. The Code Red worm [75] exploited a buffer overflow in Microsoft's Internet Information Service to execute malicious code. In the attack dubbed "Operation Aurora", Chinese hackers exploited invalid pointer reference vulnerability in Internet Explorer to compromise systems of more than 30 companies [112].

A buffer overflow occurs when the size of data written to the buffer exceeds the memory space allocated for the buffer. Without proper protection (i.e., bounds check), the excess data is written in the memory space beyond the boundary of the buffer, leading to unexpected results. Invalid pointer reference occurs when a pointer points to an object which has been deleted. If the memory for the object is allocated to another object. and if the program reads from or writes to this memory, it will result in unpredictable behavior.

The attacker may overwrite values of variables in order to alter program results, or execute his own malicious code by manipulating program data, such as return addresses or function pointers. In particular, he may execute a shell code that starts a command shell, from which he can take control of the target computer.

One typical attack is stack smashing [117], which exploits buffer overflow in the execution stack. It makes use of the fact that the return address of a function is stored in the same stack with local variables. If data is written to exceed the bounds of the memory allocated for the local variables, the memory address that stores the return address for the function will be overwritten, changing the control flow of the program. The return address will be the start address of the malicious code that the attacker provides.

A simple technique to defeat stack smashing was to make the stack space non-executable. However, hackers have found a way around by inventing the return-to- libc attack [118]. In the return-to-libc attack, the return address is replaced by a pre-existing function, rather than the attacker's malicious code. The attacker also writes data to the stack providing arguments for the function. Typical function called is the system() in the libc library, which executes an arbitrary program.

(2) Code injection:
The code injection attack is the most common way of attacking Web servers. Code is injected from outside for execution on the target computer. SQL injection and cross-site scripting are common methods used. In 2009, cross-site scripting was the most common vulnerability, followed by SQL injection [119].

Cross-site scripting [120] allows the attacker’s scripts to run on the victim’s Web browser. The primary purpose of cross-site scripting is to steal cookies of other users. For example, the attacker may insert a script, rather than text, in a message, and post the message to a Bulletin board. When a victim views the message, the victim’s Web browser executes the script and sends the victim’s cookie to the attacker’s Web server.

SQL injection [122] is a technique for embedding SQL statements in a form for submitting user inputs. The attack can succeed if the Web server does not perform proper input validation. A naı ̈ve Web server receives a user input in a form and generates a SQL statement, such as “SELECT*FROM userDB WHERE username=‘inputdata’ ”, where input data is the user input. The Web server expects, and trusts, that the user will input only alpha-numeric characters.

However, the attacker can include SQL code in the user input for execution by the Web server. The SQL code accesses sensitive records, such as passwords or credit card numbers, or damage the Web server database by deleting data. Albert Gonzalez and two accomplices stole the credit and debit card numbers of various companies, using a SQL injection attack on the payment processing systems the companies used [122,123].

(3) Cross-site request forgery:
The cross-site request forgery attack [124] is the opposite of cross-site scripting. In cross-site scripting attack, the user (i.e., the victim) regards the Web server as trustworthy. In cross-site request forgery attack, the Web server trusts its users. The attacker posts an image containing a script to a bulletin board or sends the image via e-mail, luring the victim to click on the image. When the victim clicks on the image, the script is executed and accesses the Web site where the victim is registered as an authorized user. When the victim is logged on to the Web site, the attacker can conduct malicious actions with the victim’s authority.

The Web server cannot determine if the transaction was initiated by a legitimate user or an attacker. Since the authorized user sends requests to the Web server in a cross-site request forgery attack, it is difficult to trace the attacker, or even discover any evidence of the attack. In February 2008, about 18 million users in South Korea lost their personal information because of a crosssite request forgery attack on the Auction Web site [125].

(4) Directory traversal:
In a directory traversal attack, the attacker inserts multiple “../” s to a file name that the server is to receive, so as to climb the directory structure to access files that he ordinarily is not authorized to access. The attacker aims to gain access to sensitive data, such as the password stored in /etc./passwd. The Nimda virus [126], which appeared in 2001, exploited the directory traversal vulnerability in the Microsoft Internet Information Server.

3.3.2. COMBATING HACKING

It is very difficult, if not impossible, to prevent hacking. Often it does not seem to be possible even to detect hacking. There are simple reasons for this. It is nearly impossible to have everyone create difficult-to-guess passwords, and periodically change them.

It is also nearly impossible to force everyone not to write down his passwords on a piece of paper and post it prominently. Moreover, it is just about impossible to completely avoid creating flaws in complex software. The best way to minimize vulnerabilities in software and prevent attacks is to follow best practices in the design, coding, testing, operation and maintenance of software. This is easy to say, but very difficult to do in practice, with tight deadlines, manpower shortage, lack of training and discipline in software engineering, inadequate tools, etc. Further, even if developers release software patches that fix known flaws, users or system administrators may not apply the patches, leaving the system vulnerable to attacks.

In the last 40 years we have seen a catch-up game between hackers and those who thwart them. As systems become more secure and harder to break in, it has motivated hackers to “rise to the occasion” and invent and refine hacking techniques.

Techniques have been developed to mitigate the damages caused by hacking. To defend against buffer overflow attacks, a detection technique using “canaries” has been developed [127]. A canary word is a known value that is deliberately inserted to detect buffer overflows. (The word “canary” comes from the use of canaries in coal mines to detect dangerous gas.) Fig. 3 shows how a canary word is used. The attacker tries to alter the return address in a stack frame, so that the current process is redirected to the address where the attack code is stored.

To detect this, a canary word is inserted between the memory space for local variables and the return address. The attacker would change what he thinks is the return address; however, what he ends up changing would be the canary word. By checking the canary word, one can detect if a buffer overflow has occurred. This technique has been adopted in the GCC compiler as a compile option [128].

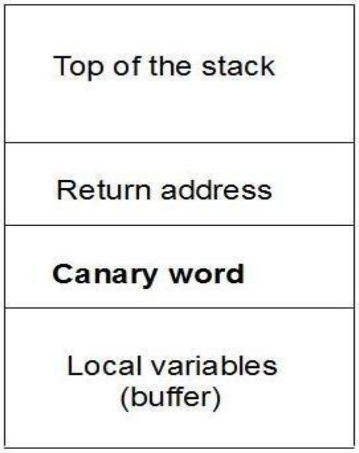


Fig. 3. Use of a canary word for buffer overflows protection.

Executable space protection technique is used to prevent code loaded into certain memory region (e.g., the stack) from executing.

Address Space Layout Randomization (ASLR) [129], included in Windows Vista and Windows 7 as a security feature, and randomizes addresses of important data areas, such as libraries, stack, and heap. The hacker cannot predict the location of the malicious code to which he wants to redirect the program. This makes the ‘‘return-to- libc’’ attack difficult to succeed.
Techniques to defend against code injection attacks also include input filtering and output escaping.

Input filtering filters out any part of the user input that can possibly be misused. For example, when the input is a user ID or a password, it filters out special characters, such as single quotes, that can be misused when SQL statements are generated from these inputs. Output escaping changes user input if part of the user input is used as output. For example, suppose a bulletin board allows users to use HTML tags when posting a message.

Then, the bulletin board must be careful not to output the message in a raw form, because a malicious user may have inserted a script (using oscript4 tag) to be executed when a reader opens the message (code injection attack). Output escaping changes oscript4 to &lt;script&gt;, so that the HTML tag is not executed on the Web browser.

Cross-site request forgery attacks are mitigated by having the server require the client to send a secret, user-specific token with form submissions. This technique makes it difficult for the attacker to forge requests, because he does not know what the user-specific token is.

There are also tools that help detect vulnerabilities in software, so that developers can analyze and remove them before they are discovered by hackers. Such tools as Nessus [89] and Retina [90] detect potential vulnerabilities in operating systems by conducting port scans and trying various exploits.

There are also vulnerability scanners for Web servers, such as Web Scarab [91], that test the Web server in order to detect if there is any chance of SQL injection or cross-site scripting attacks. Intrusion detection and prevention systems [92] can help detect and prevent hacker break-ins by monitoring network and system activities, and raising alarms if there is any suspicious activity. Intrusion detection methods include signature-based detection and anomaly-based detection. Signature-based detection makes use of known attack patterns.

The intrusion detection system suspects there is an intrusion attempt if the pattern of system or network activities matches one of the known attack patterns. Anomaly- based detection makes use of performance statistics, such as network traffic. A threshold is set which defines normal behavior, and if the performance goes beyond the threshold, an alarm is raised. Intrusion detection systems have limitations.

For signature-based detection, it is rather easy for attackers to make variations in attack behaviors to avoid detection. Further, signatures should be constantly updated to catch up with newly emerging attack patterns. For anomaly-based detection, false alarms may frequently occur, since normal behavior can change. If false alarms occur frequently, the network administrators may ignore the alarm even when a real attack occurs.

3.4. DENIAL OF SERVICE ATTACKS

There are operations, such as session establishment requests that a system must perform on behalf of all users, including potential attackers. The attacker exploits this fact and sends a large number of requests to the target system, making the system too busy to serve request from legitimate users. The targets of denial of service (DoS) attacks are not limited to end systems. They also include core network routers, switches and domain name servers.

DoS attacks came to the public's attention in February 2000, when commercial Web sites of companies including Yahoo, Amazon, and eBay were unable to service its customers for hours, resulting in damages worth US$1.7 billion [93]. The attacks were carried out by a Canadian teenager, and involved many zombies. DoS attacks were also used to bring down government Web sites.

In 2007, Web sites of the Parliament, banks and broadcast agencies in Estonia became victims of DoS attacks. In 2008, numerous Web sites in Georgia were brought down by DoS attacks. These countries were having political differences with Russia, and many suspected that Russian hackers and possibly the Russian government were behind the attacks [94].

In July 2009, a massive DoS attack was launched simultaneously on many nWeb sites of organizations in South Korea and the United States [95]. Victims included (in the United States) the White House, Secret Service, Federal Trade Commission, Transportation Department, New York Stock Exchange, NASDAQ, Yahoo Finance, Washington Post; (in Korea) the Blue House (Korean equivalent of the White House), the Ministry of Defense, National Assembly, Shinhan Bank (one of the largest banks), Chosun Ilbo (the largest circulation newspaper), and Naver (the largest portal site).

It was reported that the attacks were launched from more than 20,000 zombies. Then in August 2009, DoS attacks took place against US-based social networking sites Twitter, Facebook, LiveJournal, and Google blogging [96].

3.4.1. HOW DOS WORKS

A DoS attack is often launched from multiple sources, possibly thousands of computers; this is called a DdoS (distributed denial of service) attack. DDoS attacks make use of zombies, which send requests to a target system on command from the attacker. Often DoS attacks involve spoofing of the attackers' IP addresses as the victims' IP addresses, making it difficult to identify the attackers. Further, DoS attacks often require replies from the victim, forcing the victim to be further loaded. DoS attacks come in many forms [97–99]. Below we summarize major forms of attack.

3.4.1.1. PING FLOOD.

A ping flood is the most basic form of DoS. The attacker simply sends a large number of ping (ICMP Echo Request) packets to the target. If the target is configured to send replies, the effect is amplified.

3.4.1.2. SMURF ATTACK.

Similar to a ping flood attack, a Smurf attack uses ping packets [100]. The attacker sends ping packets, with the source IP address spoofed to be the victim’s IP address, to computers that keep a broadcast address. All computers in the broadcast address that receive the ping packet send replies to the victim’s IP address, thus attacking the victim. One packet sent to the broadcast address is amplified by the number of computers that send reply packets.

3.4.1.3. UDP FLOOD.

In a UDP (user datagram protocol) flood attack, the attacker sends a large number of UDP packets to random ports on the target. Since the UDP does not have a congestion control mechanism, the attacker can potentially send a very large number of packets. Also, the target may be configured to send the ‘‘ICMP Destination Unreachable’’ packets when there is no application listening to the port where the UDP packets arrive. The target has needless extra work to perform. This attack is generally used with IP address spoofing, so that the attacker or the zombies can avoid being detected or flooded by ICMP replies.

3.4.1.4. TCP SYN FLOOD.

TCP SYN (synchronize packet in transmission control protocol) flood [101] is a form of attack where the attacker sends a large number of SYN packets (connection requests) to the target, and fill up the connection queues on the target, so that the target cannot establish connections for legitimate TCP users. A TCP connection is established using a three-way handshake between a client and a server.

First, the client sends a SYN packet to the server. The server replies with a SYN-ACK packet and waits for the client to respond with an ACK packet. When the client sends an ACK packet, the connection is established. After sending the SYN-ACK packet, the server keeps a record of the client while waiting for the ACK packet. This takes up resource on the server. The attacker, as a client, can generate a large number of SYN packets and not send ACK packets to the server, causing the server to consume all available TCP connection queues.

3.4.1.5. DNS AMPLIFICATION ATTACK.

DNS (domain name system) amplification attack uses DNS queries [102]. The size of the reply to a DNS query can be much larger than the DNS query. The attacker creates an authoritative name server for some domain name, such as “random.com”, and registers a garbage text of large size, for example 4000 bytes, as the Text Resource Record (RR) of random.com.

After that, the attacker commands zombies to send queries to their domain name servers for the Text RR of random.com, with the zombies’ IP address spoofed to be the victim’s IP address. When the domain name servers that receive queries allow recursion, they recursively query the authoritative name server of random.com for its Text RR and get the answer. Then the domain name servers send the answer to the source IP address, which is the address of the victim. The size of the reply to a DNS query can be much larger than the DNS query.

Using a small number of zombies, this attack can overwhelm the target system.

3.4.1.6. APPLICATION-LEVEL ATTACK.

The types of attack discussed thus far all exploit network protocols or services. DoS attacks can also be carried out at the application level. For example, the attacker can command zombies to send HTTP requests to a Web server to download a large file or perform expensive database operations. This will consume CPU and network resources at the server, limiting its availability to other clients. Application-level attacks are difficult to prevent, because it is difficult to distinguish between requests from the attacker and requests from legitimate users. The attacker can carefully craft HTTP requests to make them appear as normal requests.

3.4.1.7. MAIL BOMB ATTACK.

In a mail bomb attack, the attacker sends a large number of e-mails to a target e-mail address to overflow the victim’s mailbox or slow down the mail server. The attacker may command zombies to send emails to the same target simultaneously. He may generate each e-mail with a different message to pass the spam filters.

A variant of mail bomb, called zip of death, targets mail servers that check attached files with anti-virus software. The attacker sends an e-mail with a zip file attached. The size of the attached zip file is small when compressed, but becomes huge when extracted. The anti-virus software extracts the zip file in order to check for viruses, consuming large amount of time and resources. In a popular example zip of death attack, the zip file is 42 kilobytes when compressed, but becomes 4.5 pet bytes of useless data when extracted [103].

3.4.1.8. PEER-TO-PEER ATTACK.

Conventional DDoS attacks use zombie computers to send a large amount of requests to the victim. P2P attacks use clients connected to P2P file sharing hubs. They exploit flaws in the P2P hub software to direct clients to connect to a specific IP address and a port, that is, a victim.

3.4.1.9. VARIABLE-RATE AND LOW-RATE ATTACKS.

Although it is difficult to track where the attacker is, it is generally not difficult to know when the attack actually takes place, because the server becomes unavailable or drastically slows down. The monitoring system at the target system raises an alarm when there is an abnormally large volume of traffic at a constant rate.

However, the attacker may send variable-rate and low-rate traffic to the victim, making it difficult for the victim to realize that an attack is actually taking place. Carefully crafted low-rate attack, such as the Shrew attack [104] which creates TCP packet losses and thus increases retransmission timeout for the particular TCP flow, can significantly degrade performance of the target system. If the attack is not detected, the administrators may mistakenly conclude that legitimate traffic has increased and increase investment in network bandwidth.

3.4.1.10. PERMANENT DOS ATTACK (PHLASHING).

A permanent DoS attack attempts to make hardware permanently inoperable by installing corrupted firmware. It is often called phlashing, a word that comes from flashing, which is a term used for firmware upgrade. The attacker sends false hardware upgrade requests that can bring down the hardware. Once attacked, the victim has no choice but to replace the hardware and restart the system. Network enabled embedded devices (NEEDs) are frequent targets of permanent DoS attacks, as they often have security flaws.

3.4.2. PREVENTIVE MEASURES

Many mechanisms for preventing and detecting DoS attacks have been proposed. The following are some of the measures that can be taken in order to mitigate DoS attacks

3.4.2.1. REMOVING KNOWN VULNERABILITIES IN PROTOCOL BEHAVIORS AND HOST CONFIGURATIONS.

When the ping flood packets originate from a single source without IP address spoofing, they can be easily blocked from the source address. If the attacker uses IP address spoofing or launches the attack from multiple zombies, it becomes very problematic.

The TCP SYN attack or the Smurf attack exploits vulnerabilities in protocol behaviors or host configurations. Removing such vulnerabilities can reduce the chances of DoS attacks. For example, the SYN attack exploits the fact that the server keeps half-open connections during the three-way handshake for connection establishment, and there is a limit on the number of such connections. The use of SYN cookies makes it unnecessary for the server to keep such connection states [105].

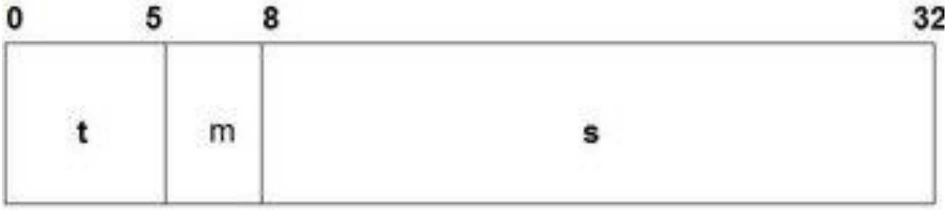


Fig. 4. SYN cookie.

A SYN cookie, shown in Fig. 4, is a sequence of numbers constructed based on the following rules:
bits 0–5: t mod 32, where t is a time stamp value.
bits 6–8: m is the maximum segment size. This would have been recorded in the SYN queue, if SYN cookies were not used.
bits 9–32: s is a cryptographic value computed based on the IP address and port number of the server and the client, as well as t. Smurf attacks can be blocked by configuring hosts not to respond to ping packets and broadcast packets. Routers can be configured not to forward broadcast packets.

Properly configuring protocols can successfully mitigate most DoS attacks that exploit protocol vulnerabilities. Although these mechanisms prevent hosts from participating in a DoS attack, they cannot prevent hosts from becoming victims of attacks.

3.4.2.2. FILTERING THE TRAFFIC.

Ingress and egress filters in the edge routers can filter suspicious packets (see Fig. 5) [106]. Ingress filtering blocks any inbound malicious packets from outside networks Ingress filters can filter out packets from a specific range of IP addresses, if the addresses have been used for DoS attacks in the past. Egress filtering blocks outbound packets originating from inside the network. One simple example is to filter out outbound packets that have source addresses outside of the address range allocated to the network inside the edge router. This can prevent attackers from generating IP-spoofed packets for DoS attacks.

Router-based filtering can filter out packets based on the source address and the network topology [107]. The router can determine whether the packets are legitimate or source-address-spoofed. These measures can mitigate DoS attacks that use spoofed IP addresses. However, recent DoS attacks do not use spoofed IP addresses. They simply make use of botnets. When the attacker packets have legitimate source IP addresses, it is difficult to distinguish the attacker packets from legitimate packets.

Naive filtering of packets may result in packets from legitimate users being dropped. Instead, a rate-limiting mechanism is used, which imposes limit on the rate of traffic sent or received on the network interface [108]. Although the attacker traffic may not be totally blocked, such a mechanism can still prevent the attacker traffic from overwhelming the host, thereby leaving room for servicing legitimate users. The rate limiting mechanism fails if the attacker uses a large botnet.

3.4.2.3. DETECTING ATTACKS.

Intrusion prevention systems can detect attacks by monitoring the behavior of the host and network [109]. They generate profiles for normal usage by analyzing system and network behaviors under normal conditions. Once they detect an abnormal behavior, they raise an alarm and invoke preventive mechanisms, such as filtering or rate-limiting. It is difficult to construct a profile for normal usage, because of the need to capture and monitor various system and network parameters on a real time basis. It is also difficult to determine the threshold and for classifying a behavior as an attack. If not done properly, there can be too many false positives and false negatives, potentially rendering the detection mechanism useless.

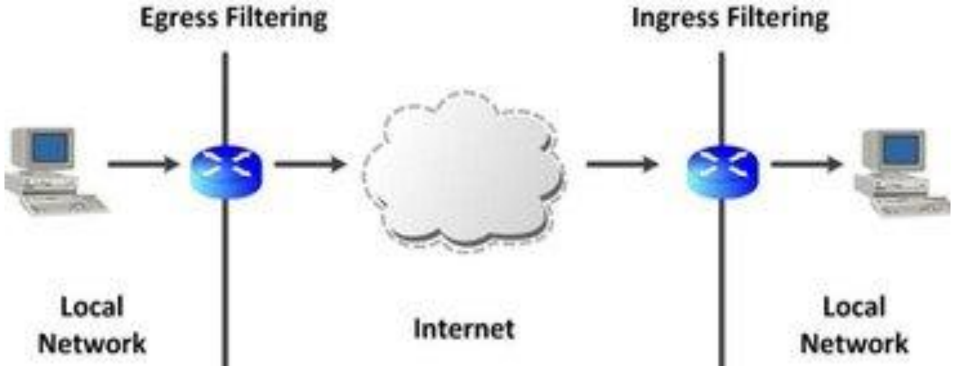


Fig. 5. Ingress and egress filtering at the edge routers.

3.5. PHISHING

The term phishing was first mentioned on the alt. Online-service. America-online Usenet newsgroup on January 2, 1996. The replacement of 'ph' for 'f' in the term phishing is probably by analogy to phreaking, which is the blend of two words 'phone' and 'freaking' to refer to the cracking of the phone network. Recently, the term pharming has also been used for online identity theft, where pharming is a play on farming.

Most methods of phishing use some plausible message in e-mail, and induce the receiver to click on a link in the email. Earlier phishing attacks broadcast the same message to all targeted people. Careful people were able to easily determine that the message was suspicious. Recent phishing attacks use more customized messages. This targeted version of phishing is called spear phishing. An attacker would gain the trust of victims by presenting information about their bidding history or shopping preferences, which are available from eBay or discoverable through their Web browser history. Several recent phishing attacks have specifically targeted senior corporate executives. This version of phishing is called whaling.

Personal profile data are bought and sold by online brokers. In 2007, hackers broke into the TD Ameritrade's database, and stole 6.3 million customers' data. Phishing attackers are increasingly targeting social networking Web sites, since they can harvest the rich trove of personal information, including the login data, personal profiles, and posts. In the second half of 2009, at least 126,697 phishing attacks were committed; this is more than double the 55,698 attacks recorded in the first half of 2009. 28,775 unique domain names were counted in these attacks.

Since one domain name can host several distinct attacks, the number of domain names is less than that of attacks.

This has decreased from 2008, while the number of registered domain names in the world has grown. Among the 28,775 phishing domains, 6372 were supposed to be registered by the scammers. The other domains were probably hacked or served on vulnerable Web hosts.

A phishing gang named Avalanche is suspected of launching two-thirds of all phishing attacks in the second half of 2009. Avalanche is regarded as a successor to another criminal operation called Rock Phish, which was rampant from 2006 into the summer of 2008. Attacks by Avalanche dwindled in 2010 [117]. The damages caused by phishing range from denial of access to online accounts to loss of money. Gartner [118] estimates that about 19 percent (11 million) of online adult users attacked by phishing had clicked on the link in a phishing attack, and 3 percent (1.78 million) provided sensitive information on forged Web sites.

Identity theft fraud generated direct losses of about US$1.2 billion in 2003. Gartner [119] estimates that losses from ATM (automatic teller machine)/debit card fraud in the United States cost US$2.75 billion, with an average loss of more than US$900. Another online survey from Gartner [120] estimates losses from phishing attacks in 2007 as US$3.2 billion. Around 3.6 million U.S. adults lost money in phishing attacks during this period. In 2008, more than 5 million U.S. consumers suffered losses from phishing attacks, a 39.8 percent increase from 2007 [121]. We note that [122] disputes these estimates as grossly exaggerated.

They claim that the victimization rate in most surveys was at most the margin of error and these questionable surveys made the situation worse by bringing a steady supply of new entrants. They estimate the annual phishing loss in the United States as US$61 million.

3.5.1. HOW IT WORKS

In order to make the receiver believe that the spoofed Web site is legitimate, the URL of the link and the Web site have to look genuine. An elementary trick used by the phishers is a misspelled URL [123]. A URL string is displayed in the status bar or the address bar of the Web browser, but people do not usually pay attention to these areas. Further, most people do not know the exact official URLs of the legitimate sites. So most people just glance at the URL and assume that it is legitimate if it looks familiar. A simpler trick is to use valid anchor text for a link, while the linked URL points to the phishers' site.

Sub domains are also used for phishing. For example, when the URL,http://www.bank.signin.com/, is shown, many people expect that the URL will take them to the signin section of a bank Web site. However, this URL refers to the bank section of the signin Web site [123]. Similarly, most people believe that http://www.bank.com@signin.com/ will lead to a page of the bank site; however, it opens the signin.com site with the username "www.bank.com"[123].

There is another problem with URLs, called IDN (international domain name) spoofing or homograph attack. In an example URL,"www.micrOsOft.com", the 0 is the number zero, but the URL is confused with"www.micro soft.com" [124]. Moreover, in the IDN system, there is a large number of code pages which look very similar to Latin character sets [125].

The open redirect is another technique to make phishers' URLs look more legitimate [126]. An open redirect is a common application that takes a URL as a parameter and redirects the user to the URL [127]. For example, the URL, "http://www.google.com/url?q=http://www.signin.com," opens a page on the signin site, not on Google.

Some phishing techniques leverage vulnerabilities in Web sites rather than spoofing them. Cross-site scripting, discussed earlier, allows the hacker to embed malicious scripts into a vulnerable Web page on the user's computer, and have the scripts executed there [128].

Recently, a new phishing technique named ‘‘tabnabbing’’ has been identified [129]. It takes advantage of the multiple tabs feature in the latest Web browsers. When the user visits a phishing page, the attacker replaces the favicon and title of the page with those of a legitimate popular Web page (e.g., the Gmail login page).

(A favicon is short for favorite icon, and is a 1616 or 3232 pixel square icon associated with a particular Web site or Web page.) If the user enters his login information, the malicious scripts direct him to the legitimate site, after sending the information to the malicious server. This happens because when the user has many tabs open, and the attacker tab is inserted into the mix, he may not realize that he had not actually opened the attacker tab.

3.5.2. PREVENTIVE MEASURES

Social response is most important to reduce losses tophishing. People can begin to avoid phishing attacks by changing their browsing habits [123]. First, they should pay more attention to the warning windows on the Web browsers. The latest Web browsers display a popup to warn against suspicious situations from negligible to serious. The problem with the warning windows is that after encountering various inconsequential warnings, people tend to start ignoring all warnings [123].

Second, people need to develop a habit to check details of the messages, and also, before responding to e-mails with sensitive personal information, contact the sender's organization to check if the e-mail is legitimate. Anti-phishing techniques have been incorporated into the Web browsers and Web site procedures [123].

Most Websites have adopted the secure authentication of Secure Sockets Layer (SSL) with strong Public Key Infrastructure cryptography. The SSL authentication is used to inform the user what the current authenticated mode is, which site the user is connected to, and which authority certifies the site.

The latest Web browsers include extensions in their address bars to display such information. Mozilla 5.0 uses an extended green favicon box (Fig. 6a) to the left of the address bar to show certificates that provide a high level of identity verification, and show other certificates with a blue favicon box (Fig. 6b). Chrome 5 uses a yellow URL bar with a lock icon (Fig. 6c) for both types of certificates.

Internet Explorer 8 fields a green URL bar for certain certificates that provide a high level of identity verification (Fig. 6d). The users are expected to understand where they intend to go [123]. However, since URLs can be complex, average users often do not know what the URL means [110].

Thus, Web browsers have been improved to help users understand where they are. Internet Explorer 8 displays the entire URL in grey, with the domain name in black. For example, in Fig. 6d, only ‘‘paypal.com’’ is in black.

It works even for sites not supplying identity verification. If the site has a high level of identity verification, an additional subwindow is attached on the right side of the address bar to show identity, such as ‘‘PayPal, Inc. [US]’’ in Fig. 6d. Mozilla 5.0 also indicates the verified identity in the extended favicon box, as shown in Fig. 6a and b. In addition, the browsers indicate who verified the identity of the site the user is connected to [123]. The problem is that not all certificate authorities (CAs) provide equally good service. Hence, a phishing site may be able to obtain a valid certificate from a vulnerable CA.

Therefore, the browser has to give the CA information to the user, who needs to be familiar with the name of the authority. The CA information can be accessed by clicking on the lock icon in the address bar or the extended favicon box. The case for Mozilla 5.0 is shown in Fig. 7. Another useful approach is to maintain a blacklist of known phishing sites and to check Websites against the list before connecting to them [123]. Internet Explorer, Mozilla Firefox, Safari, Opera and Chrome, all contain a kind of phishing-filter [111-115].

The blacklist can be used to filter out suspected phishing e-mails before they reach users’ in boxes. Spam filters specialized for phishing can prevent a fair number of suspected phishing e-mails. Another technique to prevent phishing is to divide log information into two parts. One part is owned by the user, and the other is kept on the service site. In a simple implementation, the service site asks the user to select a personal image and display the image with any forms that request his password. The user of the service site should enter his password only when he sees his image.

Thus, if the phisher asks the user for the user’s password, without the user-selected image, the user should know the request is from a phisher. According to a recent study, however, some bonehead users give away passwords in the absence of the images they had selected [116]. Mutual authentication is a protocol in which the client and server authenticate each other.

Since the server should prove its identity to the client before the client presents its certification or username/ password, this technique reduces the risk of carelessly revealing security information to the phishers. The steps in mutual authentication are shown in Fig. 8, In order to prevent phishing attacks, a strong security model has to be standardized and widely used.

However, the security model includes many participants with dissimilar interests: users, browser vendors, developers, Cas, Web server vendors, Web sites, regulators, and standards committees [123]. It is not easy to reach timely agreement among them. Security standards take many years to get ratified, adopted, and implemented. Since phishing attacks rapidly evolve, security standards may become, at least partially, obsolete, almost even before they get ratified. Further, users of the Internet who pay enough attention to the details of the look and feel of the Web pages and the URLs will be outnumbered by those who do not. When there are so many users who can be duped by the phishers, it is impossible to fully prevent or detect phishing.

3.6. CLICK FRAUD

Click fraud refers to the act of clicking on online ads for the purpose of generating a charge per click. People, automated scripts or computer programs are used to commit click fraud. Click fraud is a crime. Online advertising has become essential for promoting products, services, and causes. There are various types of online ads, including those associated with search results from search engines, display ads, and banner ads. Advertisers provide their ads, along with a budget, to ad commissioners (e.g., Google AdSense), who are brokers between advertisers and Internet publishers.

The Internet publishers are the owners of Web sites who contract with ad commissioners to display ads on their Web sites and receive payments for visitors' clicks on the ads or purchases via the publishers' Web sites. Michael Bradley claimed that he had created a software program which could allow spammers to defraud Google of millions of dollars in click fraud and demanded US$100,000 from Google for the program.

In March 2006, he was arrested for extortion and mail fraud. It is probably the first arrest related to click fraud. Yahoo, Click Forensics, and Outsell claim that in 2008, 13–17% of the clicks on search ads were fraudulent. Google disputes this, claiming that only 0.02% of the clicks were fraudulent.

Click fraud has led to some expensive lawsuits from advertisers against ad commissioners. Advertisers complained that ad commissioners, or advertising networks, did not do enough to prevent click fraud. In March 2006, Google agreed to a US$90 million settlement in a class action lawsuit filed by Lane's Gifts & Collectibles. In July 2005, Yahoo settled a class-action lawsuit, agreeing to pay the plaintiffs' legal fees, estimated at US$4.95 million. In July 2009, Facebook was sued for click fraud by RootZoo, but had a partial win.

1. Access to a protected resource is requested by a client.
2. The server's certificate is presented to the client.
3. The server's certificate is verified by the client.
4. If successful, the client's certificate is sent to the server.
5. The client's credential is verified by the server.
6. If successful, access is granted to the client.

Fig. 6. Steps in mutual authentication.

3.6.1. HOW IT WORKS

Click fraud can be committed by persons, automated scripts or computer programs which imitate legitimate users clicking on ads [154]. There are a few types of click fraud on the basis of who commits it [143]. The publisher may click on ads it hosts, possibly enlisting friends. The motivation for this type of fraud is to get larger payments from the advertisers.

A competitor of an advertiser may click on the advertiser's ads in order to drive up the cost for the advertiser or even deplete the advertiser's advertising budget. The publisher's competitor may resort to click fraud to make the publisher appear as a fraudster.

Sometimes a third party with no ties to either the publisher or the advertiser may commit click fraud. He may seek to inject confusion to the online advertising system. Often the number of invalid clicks is so small that the fraud goes unsuspected. Publishers may claim that such clicks are accidental. Click fraud on a larger scale uses a script to simulate human clicking or make visitors automatically click on ads. Botnets can also be used for click fraud. Even organized crime networks used Trojan code to create zombies and use them to generate revenue.

3.6.2. FORMS OF CLICK FRAUD AND TECHNICAL RESPONSES.

Click fraud committed by the publisher that repeatedly clicks on ads it hosts can be blocked by removing duplicate clicks that are made by the same visitor within a short period of time. The idea is very simple but, if a huge number of clicks occur continuously, it is not easy to find all duplicate clicks from the same visitors in the click stream.

Proposed an algorithm using Bloom filtering. The identification of each visitor within a time window is hashed by multiple independent hash functions, and the hash values are stored in bitmaps, as shown in Fig. 9. If all bits of the bitmaps corresponding to the hash values of a new identification are already set, the identification is regarded as a duplicate.

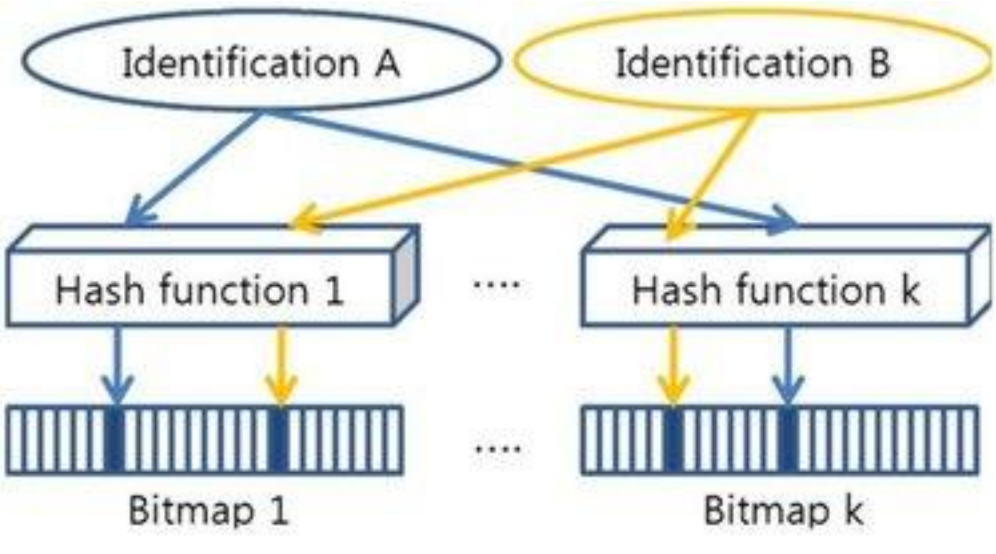


Fig. 7. Bloom filter for detecting duplicate clicks.

There are some telltale signs of potential click fraud: Keywords that normally do not get any traffic receives an unusually high number of clicks;

the number of clicks suddenly increases without having any bid changes or rank changes; a large number of clicks come from geographic regions the advertisers are not interested in. Another form of click fraud uses a script to make visitors automatically click on ads. It cannot be detected by the above method, since the clicks are from many different visitors. It is necessary to monitor the click-through rate (CTR), which is the number of clicks divided by that of impressions (i.e., ad views). It is an estimate of the probability in the ad network that an impression leads to a click.

If the CTR of a publisher is abnormally high, the publisher may be suspected. Since ads can have different probabilities depending on the nature of each publisher's Web site, abnormality cannot be simply determined by the CTR. Thus, ad commissioners often load empty ads and see whether there are clicks on them. To get around this, dishonest publishers can obtain the CTR for each of the ads for a sample period, and subsequently use the patterns proportionally to inflate the number of clicks on all the ads.

The ad commissioner may need to visit sites of suspicious publishers and check whether automatic clicks occur. Another form of click fraud uses a pair of pages (one registered page and one free page) in order to hide the automatic clicking from the ad commissioner's monitoring. The ad commissioner knows about the registered page, but not the free page. Ads are hosted on the registered page, and the free page has nothing to do with the ad.

However, the registered page has double modes of action. If a user visits the registered page directly (visitor A in Fig. 10), it behaves as a normal page and does not cause fake clicks. However, if a user visits the free page (visitor B in Fig. 10), he is automatically redirected to the registered page. The registered page recognizes that the user was redirected from the free page, and forces clicks on ads. If an ad commissioner checks the official page to determine click fraud, the page would look innocuous. To detect this form of click fraud, the pair of pages should be identified by analyzing the log records of the Internet service providers.

The pair of pages appears sequentially and close together in the log records of the ISPs. Thus, if a pair of pages has support and confidence values higher than given thresholds, the pair is likely being used for to click fraud. Another form of click fraud is to inflate the number of fake visitors by repeatedly changing their identifications, such as cookies or IP addresses. Since repeated clicks from the same identification can be relatively easily detected, the attackers try to repeatedly change the identifications of the fake visitors.

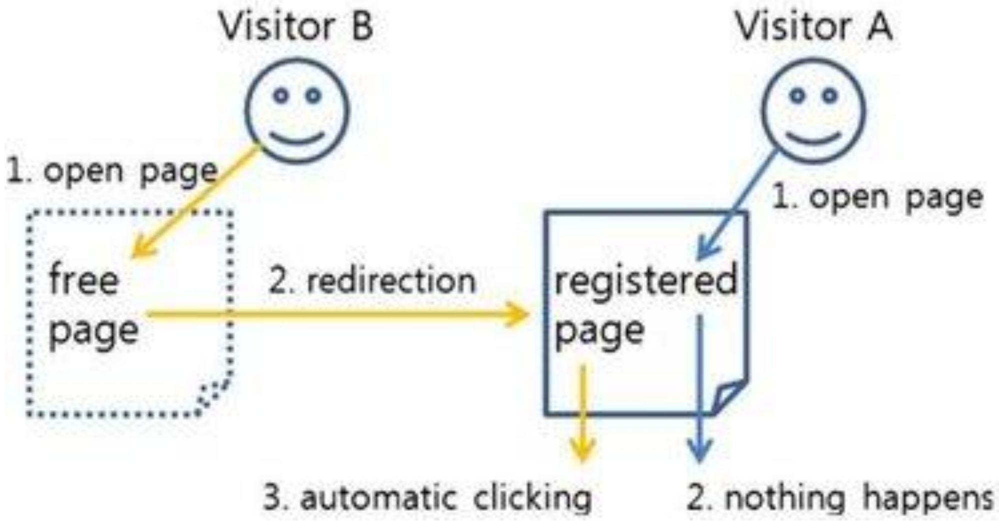


Fig. 8. Click fraud using paired pages.

Most ISPs assign a new IP address to a computer after the session initiated for the computer's connection expires. Therefore, as shown in Fig. 9, attackers can obtain new IP addresses by renewing their network connection repeatedly. The response to this form of click fraud is very similar to that for the previous form of click fraud. The probabilities that an ad hosted by a publisher is clicked on and that a visitor clicks on an ad are presumed to be independent. Thus, if these two probabilities are exceptionally dependent, it can be assumed that the visitor is controlled by the publisher [152]. As discussed earlier, the relationships between publishers and visitors can be found by analyzing the log records of the ISPs.

1. Make clicks on Web pages to increase the number of clicks.
2. Disconnect the network.
3. Wait for the session to expire.
4. Reconnect to the network to get a new IP address.
5. Return to step 1.

Fig. 9. Repeatedly obtaining new IP addresses to a computer.

The forms of click fraud discussed above are committed by a single publisher. Several fraudsters may form a coalition. Metwally et al. discusses coalition click frauds and methods to detect them. The assumption is that, if several publishers cooperate to perform a click fraud, the lists of visitors to the publishers' Web sites should be similar.

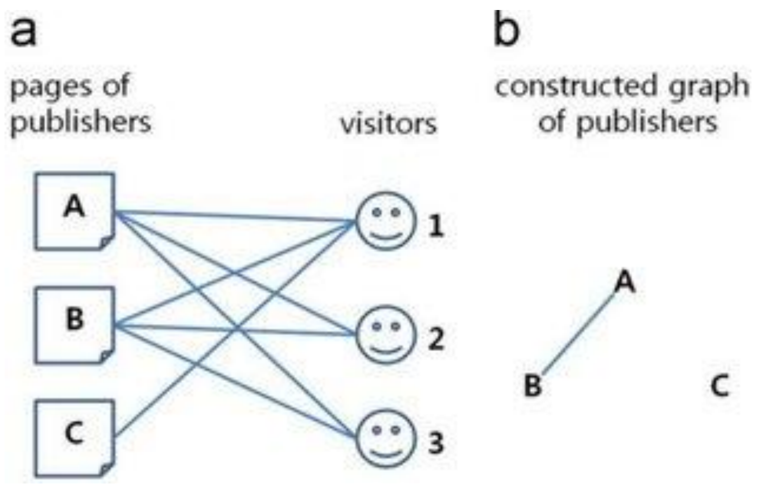


Fig. 10. (a and b) Example coalition attack.

The similarity may be measured by the set similarity of the lists of visitors. A graph can be constructed based on the similarities. In the graph, a publisher is a vertex, and an edge connects a pair of publishers if the similarity between the publishers is greater than a threshold value. Each complete subgraph of the graph is considered a coalition. For example, this Fig.10, shows the pages of publishers A and B which are clicked on by visitors 1, 2, and 3; and the page of publisher C by only visitor 1.

Then the set similarity of visitors to publishers A and B is 1.0, and that for publishers A and C, and for publishers A and B, is each 0.33. If the threshold for the set similarity is 0.5, only A and B may be connected with an edge, as shown in Fig 10. Then the complete sub graph with publishers A and B is considered a coalition. Many vendors are developing products to allow the advertisers to audit for click fraud. The products analyze the advertisers' Web server data and traffic patterns. They collect data by using JavaScript on the advertisers' Web pages and suitable tagging of the ads. Advertisers may present the results to the advertising commissioner.

New types of click fraud appear continually and often it is hard to clearly distinguish click fraud from legitimate clicks and to know who is behind a computer and what their intentions are. The Tuzhilin Report, produced as part of the click fraud lawsuit settlement between
Google and Lane's Gift, has a detailed discussion of these issues.

3.7. VIOLATION OF DIGITAL PROPERTY RIGHTS

On the Internet, file sharing has been the primary means of downloading copyrighted digital properties without authorization. File-sharing dates back to the 1980s, including Usenet, FTP servers, Internet Relay Chat, etc. Sharing of music files in mp3 format became a global phenomenon soon after Napster was released in 1999. Napster was (is) a file-sharing system that allows users to connect to a peer-to-peer (P2P) network on the Internet, search for sharable files on the computers of other users on the network, and download files of interest directly from those computers.

The original Napster was shutdown as a result of litigation from RIAA (Recording Industry Association of America). But there are many P2P file-sharing networks today, including Bit Torrent, Limeware, Shareaza, iMesh, Bearshare Lite, eMule, KCeasy, Ares Galaxy, etc.

In the 1990s, thousands of people joined some of the Internet relay chat room networks and some of the warez groups to participate in cyber piracy. Some of the Internet Relay Chat room networks, with estimated 500,000 users, were the hangouts for digital smugglers, pirates, and hunters. The members shared pirated movies, software, games, and music. They also shared codes to launch malware, and denial of service attacks. Cyber-piracy warez groups included Razor1911, DrinkorDie, FairLight, Pirates with Attitude, Class, MYTH, RTS (Request To Send), RiSC, etc. DrinkorDie may illustrate how many of these groups operated. It was started by two people in Moscow in 1993.

It gained notoriety by releasing Windows 95 on the Internet two weeks before Microsoft's official release, and releasing DVD Speed Ripper in 1999 before DeCSS (de-Content Scrambling Systems). It had zero revenue. Its organization consisted of two leaders, software and media theft specialists, a botmaster, a tester (who, "conscientiously", tested pirated software before release), and an e-mail server administrator. Based on "performance", its members had different ranks: member, senior member, and council member. With the arrival of CDs, digitization of music, CD burners, the Internet and file-sharing sites, people started to rip music off CDs and store it on PCs, and share it with anyone on the Internet without paying anything to the music industry.

Having seen sales decline significantly, the music industry made what turned out to be a controversial decision to protect music files with digital rights management (DRM) technology to prevent unauthorized downloading and distribution. In the offline world, the artists for, say, a music CD normally receives royalty payments from the music label, on the basis of the number of the original copies of the music CD sold and number of times the songs on the CD played (on radio, television, movies, etc.) and reported to the music label. In many parts of the world, the music CD may be illegally copied in full and sold, and the music label and artists do not get to collect payments due them. The buyer of the CD may play the CD anywhere as many times as he wishes, may listen to it with friends, lend it to friends, and may even make copies of certain parts of the CD and give them to friends. However, the music industry drew a line on online music.

3.7.1. HOW DRM WORKS

DRM technology refers to any technology-centric scheme that controls access to copyrighted digital content. It must satisfy two requirements simultaneously. First, it must protect the rights holders. Second it must afford the buyers legal fair use. drm-how gives some simple but good examples of DRM technology. A company sets its servers to block the forwarding of sensitive e-mail. An e-book server restricts access to, copying, or printing of material based on constraints set by the copyright holder.

A movie studio includes software on its DVDs that limits the number of copies a user can make. A music label releases titles on a type of CD that includes bits of information intended to confuse ripping software. First-generation DRM technology tried to only control copying. Second-generation DRM technology aims to control viewing, copying, printing, altering and everything else people can possibly do with digital content.

It includes three principal functions: establishing a copyright for content, managing the distribution of the content, and controlling what a consumer can do with the content.

DRM software has to define three entities – the user, the content, and the usage rights – and the relationship among them gives an informative discussion about some of the DRM techniques in use or proposed. For the convenience of the reader, we summarize some of them here. Recent software copy protection mechanisms, such as Macro vision's Safe Cast and Microsoft Product Activation for Windows and Office, require a central server to allow the installation and running of DRM software.

DRM systems for multimedia content often encrypt the content in such a way that only a particular device can play or retrieve it, enforcing constraints on how it is used. For example, DVDs use encryption to prevent users from viewing on unauthorized players. Some DRM systems use digital watermarks, a form of steganography, to record the copyright holder, the distributor, the distribution chain, and/or the purchaser. Hardware or software can use a watermark to decide whether to process the data or trace piracy. Some DRM systems are designed to confuse or disable unauthorized devices.

For example, some audio-CD copy protection systems intentionally create a faulty disk. Most CD players ignore the faults, but computers do not. Some DRM schemes make it impossible to tap into an unencrypted stream of data, even one that flows from the user's hard drive (or the Internet) to the user's display and speaker. An example is Microsoft Secure Audio Path, which embeds DRM within the Windows operating system.

3.7.2. DRM TECHNOLOGY

DRM technology has different manifestations depending on the types of content it aims to protect and devices on which the content is recorded. [163] provides a good summary of DRM technologies for five types of content: film, television programs, music, games, and e-books. For the convenience of the reader, we summarize it below. An early DRM technology used on film DVS was the Content Scrambling System (CSS) from the DVDForum. The CSS encrypted the content.

Microsoft's Windows Vista includes a DRM technology called the Protected Media Path, which in turn contains the Protected Video Path (PVP), also an encryption-based technology. Advanced Access Content System (AACS) is a DRM technology for HD DVD and Blue-ray disks. AACS was developed by AACS Licensing Administrator, a consortium that includes Disney, Warner Brothers, Toshiba, Sony, Microsoft, IBM, and Intel.

The Cable Card standard is used by cable television providers in the United States to restrict content to services to which the customers subscribe. The Content Protection and Copy Management (CPCM) technology is being developed by DVB(Digital Video Broadcasting)-CPCM, an industry led consortium of around 250 broadcasters, manufacturers, network operators, software developers, regulatory bodies and others in over 35 countries to design open technical standards for the global delivery of digital television and data services.

CPCM is to specify a way of adding information to digital content, such as television programs, to describe how and if content may be used and shared among other CPCM-enabled devices. Content providers can use a range of flags stored with the content to describe how it may be used.

All CPCM-enabled devices should obey these flags. These flags can allow or deny content to be either moved or copied to other CPCM devices. Content may also be provided for a set time limit, or forbid content to be played concurrently on separate devices.

In 2002, Bertelsmann first used DRM on audio CDs. In 2005, Sony introduced a new DRM technology which installed DRM software on users' computers without their being aware. Many online music stores employ DRM to restrict usage of online music downloaded. There are many options for consumers, including iTunes, run by Apple Computers; Napster; Wal-Mart, Sony, etc. Although DRM is widely used for Internet music, some online music stores, such as eMusic, Dogmazic, Amazon, Beatport, etc., do not use DRM.

Computer games sometimes use DRM technology to limit the number of computers and devices on which the games may be installed by requiring online authentication via the Internet. Most games allow three to five installs; some allow an install to be recovered when the game is uninstalled. The two most prominent DRM technologies in use are SecuROM and Steam. E-book readers use DRM technology to limit copying, printing and sharing. Amazon Kindle, Microsoft Reader, Adobe Reader, Ereader are some of the popular e-book readers on the market.

Microsoft Reader includes a digital ID for the purchaser of the book, making it possible to trace the purchaser, and thus discouraging distribution of the e-book. Similarly, Ereader from Palm links the credit card information of the purchaser to the e-book copy purchased. In July 2009, Amazon remotely deleted copies of George Orwell's 1984 and Animal Farm from customers' Kindle readers, and credited their accounts for the price.

Amazon had little choice, since the publisher changed its mind about offering an electronic edition. Someone complained that it was like an employee of the Barnes & Noble bookstore sneaking into a customer's home in the middle of the night, taking some books the customer had purchased, and leaving a check on the coffee table for the books. This story raises an issue about the extent of control a vendor has on online purchases of digital properties.

3.7.3. DRM CONTROVERSY

DRM technology places a limit on the types of devices (desktop, laptop, netbook, MP3 player, etc.) and the number of devices on which the buyer may transfer the digital property and play, and on the number of friends with whom he may share the property. This is where the legal fair use becomes problematic for the buyer. Many users, angered by the DRM restrictions on games, sought pirated copies of the games, and even filed lawsuits against the game publishers.

With no qualms, consumers have sought out, used and disseminated pirated copies of the games. In 1999, Jon Lech Johansen, a Norwegian teenager, released a DeCSS that allowed CSS-encrypted DVDs to play on computers running Linux. In December 2006, a process key was published on the Internet by hackers, enabling unrestricted access to AACS-restricted HD DVD content.

After the cracked keys were revoked, further cracked keys were released. Many other hackers have successfully defeated most, if not all, restrictions erected by the DRM barriers.

They and others have similarly posted “how to” information to the Internet to “help” ‘‘friends’’ who like to listen to music free of charge. The Sony DRM on audio CDs was a disaster. It surreptitiously installed software, which included a rootkit. (A rootkit is malware that enables continued privileged access to a computer. It can target the BIOS, boot loader, kernel, hypervisor, etc.) Sony had to recall millions of CDs and released patches to remove the rootkit from the software its DRM installed. Sony’s DRM could be trivially bypassed by holding down the ‘‘shift’’ key while inserting the CD or by disabling the autorun feature. Further, audio tracks could be played and re-recorded, thus bypassing the DRM entirely. In 2007, EMI stopped publishing audio CDs with DRM.

The four major music labels (Universal, Sony BMG, EMI, and Warner) no longer release audio CDs with DRM. Further, many major online distributors of music have chosen to release music without DRM. There are two fundamental problems to DRM technologies. One is the difficulty of satisfying the legal fair use requirement in the face of the wide variety of devices on which people may play digital properties. Another is the mindset of the people who have become accustomed to receiving valuable contents and services for free on the Internet.

3.8 NON-TECHNOLOGY-CENTRIC RESPONSES TO THE DARK SIDE

There are at least six types of response, besides technology, to addressing the dark side of the Internet: legislation, law enforcement, litigation (by the government, industry, and individuals), international collaboration, actions by volunteers, education of the Internet users (by the governments and volunteer groups), and awareness and caution by everyone. Since any effort to address a problem is made necessarily after the problem becomes serious, and new ways of misusing and abusing the Internet are continually created, any approach to address the dark side enters a perpetual catch-up cycle.

3.8.1. LEGISLATION

Many countries have enacted legislation in their fight against spam, dissemination of malware, hacking, denial of service attacks, phishing, click fraud, violation of digital property rights, violation of privacy, etc. However, like technology, legislation has not kept pace with the rapid evolution and spread of the online offenses. To make matters worse, usually it takes at least a few years – a very long time in the age of the Internet – to create new laws or revise existing laws. Nonetheless, the laws are necessary as a basis for prosecuting those cyber crooks who do get caught.

Today, laws regarding cyber bullying, online gambling, digital property rights and even spam are inadequate. Lorie Drew, the mother who drove Megan Meier to suicide was initially convicted on the grounds that she had violated the terms of use of her MySpace account by falsifying her personal profile. The conviction was later overturned, and she went free [53]. After the suicide of Jinsil Choi, the South Korean actress, although some of the people responsible for disseminating the rumors were identified, none of them was convicted of any crime. Some of them did quit their jobs because of bad publicity.

Many countries have laws against spamming. For example, the European Union passed Article 13 of the EU Directive on Privacy and Electronic Communications in 2002. Australia adopted Spam Act in 2003. The United States adopted the Can-Spam (Controlling the Assault of Non-Solicited Pornography and Marketing) Act in 2003.

The Can-Spam Act, in particular, sets the rules for commercial e-mail, establishes requirements for commercial messages, gives recipients the right to have others stop e-mailing them, and spells out tough penalties for violations. Each separate e-mail in violation of the Can-Spam Act is subject to penalties of up to US$16,000. The law prohibits use of false or misleading header information and deceptive subject lines. It requires the sender to identify an advertising e-mail explicitly as an ad, tell recipients how to opt out of receiving future e-mails from the sender, honor opt-out requests promptly, etc.

The problem with anti-spam laws is that they are totally ignored by the spammers. In fact, some estimate that 63% of the opt-out option does not work. When recipients click on the ‘‘unsubscribe/remove me’’ button on the spam e-mails, often they merely end up confirming to the spammers that they (the spammers) reached valid e-mail addresses, and so end up receiving a lot more spam e-mails later.

Similarly, the digital property rights laws on Internet distributed music have been scoffed at or circumvented by tens of millions of otherwise law-abiding people. Most countries now have laws against hacking. For example, in the United States, the Computer Fraud and Abuse Act, passed in 1986 and amended several times since then, defines hacking with the intent to defraud as illegal, which can result in heavy fine or many years in jail. In the United Kingdom, the Computer Misuse Act 1990 criminalizes hacking.

An example of the anti-click-fraud legislation is the (United States) California Penal code 502(C). It is a fairly comprehensive computer data access and fraud act [168]. The pay-per-click online advertising industry is trying to propose tighter laws that will cover the issues not bound by contracts.

3.8.2. LAW ENFORCEMENT

Many governments often do not have adequate budget, or, even if they do, they do not have enough trained manpower to deal with cyber crooks. In fact, many governments do not even have laws on cyber crimes. Besides, in every country, the wheels of criminal justice turn very slowly, especially compared to the speed with which the dark side of the Internet has spread. Despite this unsatisfactory state of affairs, there have been many major arrests and crackdowns that have received wide coverage in the news media during the past several years.

They include master spammers, online pirates, operators of spam ISPs, operators of illegal file-sharing sites, data thieves, executives of illegal online gambling sites, hackers, DoS attackers, malware creators, fraudsters, child pornographers, etc. These cases suggest that there have been many more, less spectacular, cases of prosecutions and disruptions of cyber crooks. Below we highlight some of the major crackdowns that have occurred and covered by the main-stream news media in the United States during the past several years.

The United States has aggressively gone after master spammers, and successfully prosecuted many of them.

Spam kings include Alan Ralsky and Robert Soloway. Ralsky sent 70 million spam e-mails a day; in 2009, he received a 51-month sentence in US federal prison. In 2008, Soloway received a 47-month sentence in US federal prison. In 2008, the Federal Trade Commission shutdown a spam ring referred to as HerbalKing, which promoted replica watches and a variety of pharmaceuticals. The ring includes accomplices in the United States, Australia, New Zealand, India and China.

They shipped drugs out of India, had its Web sites in China, and processed credit cards from Georgia and Cyprus. They used a botnet made up of 35,000 computers to send 10 billion spam e-mails a day. They made US$400,000 a month. In 2010, Albert Gonzalez was sentenced to 20 years in US federal prison for masterminding, with two Russian accomplices, the theft and reselling of 130 million credit and debit card numbers between 2006 and 2008 from various companies, including 7-Eleven, Heartland Payment Systems, Hannaford Brothers, and two unnamed national retailers. In 2009, the United States also cracked down on a phishing ring.

The ring involved 53 people from the United States and Egypt. They created fake Bank of America and Wells Fargo Web sites, and from Egypt sent mass e-mails. The leaders in the United States, after receiving the necessary customer data from Egypt, stole money from the victims' accounts. The ring stole at least US$2 million in two and a half years. In 2005, Brazilian police arrested Valdir Paulo de Almeida for leading an 18-member phishing ring, which stole up to US$37 million in a 2-year span. The ring spammed approximately 3 million e-mail accounts on a daily basis. The e-mails included a Trojan, which included a keystroke logger that captured user names and passwords, and sent them back to the phishing ring.

In 2006 Japanese police arrested eight people for phishing by spoofing Yahoo Japan Web sites, and stealing US$870,000. In 2004, the US FBI (Federal Bureau of Investigations) arrested several people associated with the Gambino mafia on the largest Internet fraud, pornography and telephone scams in U.S. history—over US$400 million. They purchased a bank and a telephone company to carry out the fraud.

They advertised free tours of porno sites, and, using cramming (discussed in Section 3), billed the unsuspecting viewers for bogus phone services [174]. In December 2001, the United States, the United Kingdom, Australia, Sweden, Finland, Norway and Germany launched global raids, code named Operation Buccaneer, to shutdown most of the warez group active at that time, including Razor1911, DrinkorDie, RiSC, popz, etc.

The United States made online gambling illegal in 2006, and decided to disrupt illegal online gambling by aggressively going after the executives. The laws they invoked have included anti-racketeering laws; anti-gambling laws of some state; laws on wire wager, tax fraud, money laundering, etc. These actions, some of which did not hold up in the courts, nonetheless frightened investors into selling their shares in the businesses, and frightened the executives of the businesses into resigning or even shutting down businesses.

In July 2006, US FBI and IRS (Internal Revenue Service) arrested David Carruthers at Dallas-Fort Worth international airport en route to Costa Rica from the UK. Carruthers was CEO of BetonSports, an online gambling business based in Costa Rica. He was placed in house arrest in a hotel and had to wait for trial for more than 3 years. He was sentenced to 33 months in federal prison for violating the US RICO statute (”Racketeer Influenced and Corrupt Organizations Act”) for accepting bets from US citizens via telephone or the Internet.

Gary Kaplan, founder of BetonSports, received a 51-months prison sentence, and fined US$43 million, while Beton- Sports was fined US$28 million. BetonSports went out of existence. In September 2006, the US Customs arrested Peter Dicks at New York City’s John F. Kennedy International Airport. Dicks, a British citizen, was chairman of Sportingbet PLC which operates online gambling business.

He was arrested under an outstanding warrant issued by the state of Louisiana that charged Mr. Dicks with running a gambling enterprise by computer, a crime under Louisiana law. A New York judge ruled that the warrant was not enforceable in New York, and Dicks was free to return to the UK. The basis of the ruling was that Internet gambling is not a crime in New York, and so New York does not have the authority to extradite Dicks to Louisiana. However, Dicks resigned as chairman of Sportingbet.

3.8.3. LITIGATION

The music industry has been hurt badly by the proliferation of free file-sharing programs that allowed millions of people to share, free of charge, music files via the Internet. The US music industry, represented by the RIAA (Recording Industry Association of America), has resorted to litigations. The RIAA sued illegal file-sharing sites, including Napster, Aimster, Grokster, AudioGalaxy, etc. Those sites either went out of operation or now operate legally, paying royalties.

The RIAA has even sued individuals who downloaded and distributed music files. Some estimate that the RIAA has gone after about 35,000 individuals for up to US$150,000 per song, and many of the defendants settled for around US$3000–5000 [179]. Joel Tenenbaum, a Boston University student, contested the lawsuit and, perhaps because of strategic mistake by his lawyer, Charles Nesson, a Harvard law school professor, lost. He ended up with a US$675,000 fine, for downloading and distributing 30 songs.

3.8.4. INTERNATIONAL COLLABORATION

Since cyber crimes cross national boundaries, international collaboration and international agreements are essential to defeat them. International collaborations have indeed taken place to crack down on egregious cyber crooks. As discussed, many cyber-piracy groups and Internet Relay Chat room networks were shutdown as a result of international collaborations. Many countries cooperate to halt major malware and denial of service attacks, and capture those responsible for initiating and amplifying the attacks.

However, there are some fundamental limits to international collaborations. Spammers and other cyber criminals have typically used open relay servers, open proxy servers, and zombies located in countries outside their residence to avoid discovery and prosecution. Many have been successful in evading prosecution over long periods.

There are several reasons for the limits on international collaborations.

1. Different countries have different laws on cyber crimes; in fact, some countries have no laws. Some countries punish cyber crooks with long jail terms, while others apply only a slap on the wrist. For example, in the United States, online gambling is illegal; however, in the United Kingdom, France and some other countries, it is not. As another example, some countries ban social Web sites that allow pornographic materials, while others do not. Some ban social Web sites that are used by people to mobilize protesters against the governments, while others do not.

2. It takes a long time to change existing laws, and possibly even longer to create new laws, in order to participate in international collaboration.

3. Different countries have different law enforcement capacities. In fact, most, if not all, countries have been losing the war on cyber crimes.

4. People of different countries abide by the law and ethics to different degrees. As an example, Onel de Guzman, a young Filipino man who created and disseminated the “Love Bug” virus in 2000 and caused billions of dollars in damage around the world, was hailed by the public as a national hero in the Philippines, for “having shown the world the technical prowess of the Filipinos”.

5. Some countries simply do not like or trust some other countries to start with, and not much cooperation can be expected on such “minor” issues as “cross-border cyber crimes”.

6. Some countries see no need to prosecute cyber crooks when the cyber crooks inflict damage to people and computers in other countries.

7. Comprehensive international agreements on cyber crimes are a long way off. The Convention on Cyber Crime [181] is an international treaty signed or ratified by 46 countries since 2001. Even if not signed or ratified, many countries consult the convention when drafting their law on cyber crimes. However, rather than joining the convention, developing countries want a new treaty in which they can participate from initial drafting. Russia, backed by developing countries, proposed a United Nations treaty on cyber crime, but was rejected in April 2010 due to oppositions from the United States and the EU countries.

3.8.5. VOLUNTEERS ACTIONS

There are numerous volunteer groups that try to help defeat cyber crimes. They may be classified into four categories: those that gather and provide information, those that develop technology products for addressing cyber crimes, those that advocate the creation of laws, and those that actively go after cyber crooks. The following are examples of groups that provide information to people to help them guard against cyber crimes.

The Spamhaus Project (http://www.spamhaus.org) tracks e-mail spammers and spam-related activities. spamhaus is a pseudo-German expression for an ISP or other firm which spams or willingly provides service to spammers.

Anonymous Postmasters Early Warning System (http://www.apews.org) is a list of areas on the Internet which several system administrators, ISP postmasters, and other service providers have assembled and use to deny e-mail and in some cases, all network traffic from. The Shadow server Foundation (http://www.shadowerver.org) is a volunteer group of professional Internet security workers that tracks and reports on malware, botnet activities, zombies, attackers, and electronic fraud.

Anti-Phishing Working Group (http://www.antiphishing.org/) is a global industrial and law enforcement association focused on fighting fraud and identity theft that result from phishing. PhishTank (http://www.phistank.com/) is a clearing house for information about phishing. It provides an open API for developers and researchers to integrate anti-phishing data into their applications. Millersmiles (http://www.millersmiles.co.uk) launched the world’s first scam alert service using an RSS news feed. The news feed has been used to fight phishing based identity theft.

Honeypots (Research honeypots) (http://www.honeynet.org) gathers information about the motives and tactics of the hackers targeting different networks. The honeypots are used to research the threats that organizations face, and to learn how to better protect against those threats. Research honeypots capture extensive information, and are used primarily by research, military, or government organizations.

StopCyberBullying.org (http://www.stopcyberbullying.org) aims to help parents and teachers help children and students cope with cyber bullying, with advice on how to stop it and what action to take. The following are examples of groups that develop technology responses to cyber crimes:
Conficker Working Group is an organization formed to understand and eradicate the Conficker malware. It consists of security researchers from more than two dozen Internet, software and security vendors, university researchers, and law enforcement officials.

Stop Badware (http://www.stopbadware.org) aims to fight malware. The organization is run by the Berkman Center for Internet and Society at Harvard Law School, and Oxford University’s Oxford Internet Institute. Support is being provided by Google, Mozilla, Lenovo, PayPal, VeriSign, and Sun Microsystems. Consumer Reports Web Watch is serving as an unpaid special advisor.

The National Anti-hacking Group (http://www.nag.co.in) consists of ethical hackers and cyber security experts, and aims to reduce cyber crimes by raising awareness of cyber security. The organization also does research to discover vulnerabilities in Web sites and networks, and develop defensive measures to overcome exploits of these vulnerabilities.

SOSDG (The Summit Open Source Development Group, (http://www.sosdg.org) promotes the development of open-source software for fighting spam methods. The following is an example of groups that advocate the creation and adoption of laws against cyber crimes:

CAUCE (The Canadian Coalition against Unsolicited Commercial E-mail, (http://www.cauce.org) is a consumer advocacy organization against spam e-mail. It advocates the creation and adoption of anti-spam laws. Some volunteer groups actively go after cyber crooks.

Groups, such as ‘‘Bust Auction Scams’’, are vigilantes that go after those who do not pay the sellers of items they bought on Internet auctions. Some groups, such as ‘‘Perverted-Justice.- com’’ and ‘‘Counter Pedophilia Investigative Unit’’, go after known convicted sex offenders on the Internet, for example, members of social networking sites.

Some of these groups may be violating the law in trying to do good deeds. Some groups, such as Unspam Technologies, file lawsuits against both the cyber crooks and the [183]. For example, by suing the banks that were victimized by hacking, such groups hope to force the banks into divulging details of the thefts, including the names of victims, etc. that they can possibly use to identify hackers. This legal method has been used by others to identify spammers, to gather information on illegal Internet pharmacies, and to force Internet service providers to name customers who were illegally sharing music files.

3.8.6. EDUCATION

Government agencies, schools, news media, businesses, and volunteer groups have made efforts to educate the public on ways to guard against most elements of the dark side of the Internet. The subjects include safe uses of the Internet, especially the social Web sites, for teenagers and their parents; warnings against joining suicide Web sites; warnings about various types of Internet and e-commerce frauds; tips on securing the PCs against malware; warnings about phishing, etc. The following are some of the groups and organizations that try to educate the public on guarding against cyber crimes.

Cyber Citizen Partnership (http://www.cybercitizenship.org) was established by the Information Technology Association of America (ITAA) and the United States Department of Justice. It provides programs for educating children and young adults on the dangers and consequences of cyber crimes. It also provides materials for parents and teachers to help them teach young people good online citizenship.

Cyber Angels (http://www.cyberangels.org), launched by the (United States) Guardian Angels in 1995, is one of the oldest online safety education programs in the world. Their first activity was to help women who were being harassed on Internet Relay Chat. Their current goal is to provide education to prevent becoming victims of cyber crimes and to assist victims in tracing and identifying the perpetrators of online crime.

GetNetWise (http://www.getnetwise.org) is a project of the Internet Education Foundation to provide materials on Internet safety to parents and educators, so that they can teach young people.
The Internet Keep Safe Coalition (http://www.ikeepsafe.org) is an organization that brings together (the United States) governors and their wives, attorneys general, public health and educational professionals, law enforcement and industry leaders to work for the health and safety of youth online. The organization provides materials for parents and educators to teach children.

Scambusters.org (http://www.scanmbusters.org) provides a public service Web site and e-mail newsletter to help people guard against Internet scams, identity theft and spam. Identity Theft Resource Center (http://www.idtheftcenter.org) provides victim and consumer support as well as public education. The ITRC also advises governmental agencies, legislators, law enforcement, and businesses about identity theft.

NCSA (US National Cyber Security Alliance) builds public/ private partnerships (staysafeonline, http//www.staysa feonline.org) to create and implement education efforts on cyber security.

3.8.7. AWARENESS AND CAUTION

Just as in the offline world, people have to be aware of the dangers in the online world, and exercise caution to avoid the dark side. For example, in the offline world, people have learned not to give out sensitive personal data lightly, not to get into deals that seem to be too good to be true, not to venture into certain parts of large cities after nightfall, not to believe everything they read and hear in the news media, etc. To avoid victimized by the dark side in the online world, people have to learn about the dangers, and exercise caution. There are many things they have to do. The following is a short list.
 People should ignore ads that offer ‘‘an opportunity to make US$10,000 a month working part-time at home’’, or something similarly generous.

People should ignore e-mail from some random person that offers ‘‘a generous share of a slush fund that a deposed Nigerian dictator had set up’’, or something similarly generous. People should not click on the ‘‘remove me’’ button on spam e-mail. They should ignore e-mail from anyone telling them that they need to confirm sensitive personal data, such as credit card or debit card number, bank account number, social security (resident) number, passport number, etc. If in doubt, they should find the phone number, and call to verify legitimacy of the confirmation request. They should not give out credit card information that free porn sites requests for ‘‘age verification purpose only’’.

When they received an e-mail from a friend that says. I’m traveling in Europe, and my bags containing the passport, airline ticket, money, and everything have been stolen. I need to buy a return ticket home. Please wire me US$2,000. I will pay back promptly after my return’’, they should assume that the friend’s e-mail account has been hacked into and that a scammer is sending out the e-mail to everyone on the friend’s address book.

They should assume that any comments, photos or videos they post to the Internet will be not only viewed by friends, but carefully examined by people such as the police, school administrators, university admissions officers, prospective employers, recruiters, etc. They should not believe everything they read or view on the Internet. There are intentional and unintentional inaccuracies in blogs, Wikipedia articles, forums/bulletin boards, etc. Some photos and videos may be hoaxes, or may be taken out of context. Some posts and blogs amplify and disseminate false information.

3.9 TESTING OF ANTI-ROOTKIT SOFTWARE FOR THE DETECTION AND REMOVAL OF ROOTKITS III.

In these days, rootkit technologies are gaining more and more popularity with virus writers. The cause for this is quite obvious: they make it possible to hide malware and its components from PC users and antivirus programs. You can find the source codes for ready-made rootkits easily in the Internet free access that inevitably brings about widespread of this technology in various Trojan software or spywares.

Rootkit (from the English root kit) is software for hiding the malefactor's or malware presence traces in the system. Rootkit technologies allow the malware to hide its activity in the victim's computer by disguising the files, processes as well as its presence in the system. A lot of specialized software products known as anti-rootkits exist for malware detecting and removing.

The motive of this book is to evaluate the ability of the most popular antivirus and anti-rootkit products to detect and remove malicious programs ('in-the-wild' samples) that use rootkit technologies and actively circulate over the Internet Wide-spread ITW malware testing gives us a good idea of how well the anti-rootkit software under analysis can cope with well-known rootkits.

It should be noted that although testing of in-the-wild malware samples is of real practical use, there is also a great deal of research value in ascertaining the capabilities of proactive detection when combating the hidden threat of rootkits. There are numerous specialized anti-rootkit products available for the detection and removal of these types of malicious programs. Furthermore, many antivirus developers state that their products include a function to detect active rootkits.

3.9.1 COMBATING MALICIOOUS SOFTWARE

If you are looking for basic technical information on how to protect the privacy of your data — whether it's on your own computer, on the wire, or in the hands of a third party. Although we hope you'll have the time to review all of the information in the SSD guide, if you're in a hurry to get to the technical details, this is where you can read articles that will explain:

- the basics of the relevant technologies, such as the Internet basics and Encryption Basics articles
- how to improve the security of different communication applications, such as your web browsers, email systems and IM clients
- how to protect your privacy by using defensive technologies such as secure deletion software, file and dish Encryption software, and virtual private networks
- the overarching security threat posed by malware, how to evaluate that threat, and how to reduce it

Just remember: technology changes quickly. We'll be doing our best to keep these articles updated to reflect current developments, but in the meantime, you should take the time to review information from multiple sources before making any serious security decisions.

These articles and notes describe my perspective on malware threats and associated defensive techniques. Join the fight by learning how to combat malicious software.

3.9.2 MALWARE SAMPLE SOURCES FOR RESEARCHERS

This list offers links to sites where malware researchers can download samples of malicious software. Combating malware in the Enterprise Course.

SANS' 2-day Combating Malware in the Enterprise course, presents a practical approach to discovering and mitigating malware threats in an enterprise environment.

5 Steps to Building a Malware Analysis Toolkit Using Free Tools

Examining the capabilities of malicious software allows your IT team to better assess the nature of a security incident, and may help prevent further infections. Here's how to set up a controlled malware analysis lab—for free.

3.9.3 ANALYZING MALICIOUS DOCUMENTS CHEAT SHEET

This cheat sheet outlines tips and tools for reverse-engineering malicious documents, such as Microsoft Office (DOC, XLS, and PPT) and Adobe Acrobat (PDF) files. What to Include in a Malware Analysis Report. This note summarizes my recommendations for what to include in the report that describes the results of the malware analysis process.

3.9.4 UPDATES FROM TWITTER USERS WHO COVER MALWARE

I maintain a list of Twitter users whose updates focus on malicious software threats and defenses. You can use this page to read the latest updates from these individuals.

3.9.5 ON-LINE TOOLS FOR MALICIOUS WEBSITE LOOKUPS

Several organizations offer free on-line tools for looking up a potentially malicious website. Some of these tools provide historical information; others examine the URL in real time to identify threats.

3.9.6 BLOCKLISTS OF SUSPECTED MALICIOUS IPS AND URLS

Several organizations maintain and publish blocklists/blacklists of IP addresses and URLs of systems and networks suspected in malicious activities on-line. This brief note mentions several such publicly-available lists.

3.9.7 AUTOMATED MALWARE ANALYSIS SERVICES

This brief note provides a listing of freely-available services that automate key behavioral analysts tasks for malicious software.

3.9.8 REVERSE-ENGINEERING CHEAT SHEAT

This is a cheat sheet of shortcuts and tips for reverse-engineering malware. It covers the general malware analysis process, as well as useful tips for OllyDbg, IDA Pro, and other tools.

3.9.9 USING VMWARE FOR MALWARE ANALYSIS

This article describes an approach to using VMware virtualization for setting up a malware analysis sandbox.

3.9.10 REVERSE-ENGINEERING MALWARE PAPER

This paper defines a framework for using easily-accessible tools and a dual-phased approach to examine malware such as viruses, worms, and Trojans.

3.9.11 THE EVOLUTION OF MALICIOUS AGENTS.

This article examines the evolution of malicious agents by analyzing popular viruses, worms, and Trojans, and detailing the possibility of a new breed of malicious software. This article describes an approach to using VMware virtualization for setting up a malware analysis sandbox. Reverse-Engineering Malware Paper. This paper defines a framework for using easily-accessible tools and a dual-phased approach to examine malware such as viruses, worms, and Trojans.

3.9.12 STOPPING MALWARE ON ITS TRACKS.

Malicious software helps attackers infiltrate network and system defenses, disrupt business operations, and funnel sensitive data out of corporate and personal computers. Unfortunately, there is no single-step fix to preventing and even detecting infections. Stopping malware requires an approach grounded in awareness and control.

This article presents recommendations for addressing the risks associated with modern malware.

3.9.13 BE ATTUNED TO THE STATE OF YOUR NETWORK AND SYSTEMS

Malicious software, such as bots and spyware, often goes unnoticed for far too long. Well-crafted malware can avoid being detected by antivirus software and intrusion detection systems. The first line of defense against such a formidable foe is to become familiar with the normal state of your IT infrastructure, and monitor it to detect anomalies. Establishing and maintaining IT infrastructure awareness means committing to the following steps:

- Centrally manage logs from systems and network devices across the enterprise to detect anomalous events. Even an operational incident, such as a surge in CPU load on a server, could have security implications; the increased load could be attributed to malware on that system. Logs can be aggregated without commercial tools via Syslog, which runs natively on Unix and has been ported to Windows. Without a central monitoring point, your perspective on the infrastructure will be severely obstructed.

- Deploy intrusion detection sensors at key points on the network. Host-based sensors on key servers also help. However, maintaining host intrusion detection systems (IDS) tends to be more burdensome than managing network IDS. Even though traditional IDS may not block infections, it will offer additional visibility into the environment. Snortis widely considered the king of free network IDS tools. For a free multi-platform host IDS, take a look at OSSEC.

- Monitor outbound network traffic to detect infected systems that seek instructions or leak data to their masters. You can tune a network IDS sensor to scrutinize outbound traffic, or employ traditional network monitoring tools for this purpose. (I have had a lot of luck with freeArgus open project software.) The quicker a compromise is detected, the faster it can be contained. To learn more about detecting unauthorized activities in outbound traffic, see the book Extrusion Detection by Richard Bejtlich.

- Detect unauthorized changes to the state of your systems. Although some malware resides purely in memory of the infected system, many infections leave footprints on the file system or registry. Some host IDS can detect such changes to the system's integrity. Free tools dedicated to accomplishing this include AIDE cfengine, and the open source version of tripwire.

3.9.14 TRAP MALWARE WITH HONEYPOTS

Honepots combine the best aspects of detective and preventative technologies in the fight against malware. Honeypots are systems specifically deployed to be compromised.

While the development of commercial honeypots seems to have lost steam, there is a plethora of innovative and freely available honeypot technologies. When carefully deployed, they can strengthen an enterprise's defensive posture in several ways:

- Slow down an intruder's progress by having him waste time breaking into a system that offers no value to the attacker. For instance, the free LaBrea _ tool stalls port scans and worm propagation activities by creatively responding to an attacker's network connections.

- Decrease the rate of false positives, which often plagues network IDS. Since a honeypot, by definition, should not participate in production activities, almost any connection to it is an indication of malice. A free tool Honeydemulates_servers, devices, and even networks to increase the span of such monitoring without requiring multiple physical systems.

- Capture malware samples for analysis. Since malware is a part of most modern intrusions, capturing it before it finds its way to a production system assists in incident response. One of the free tools that can assist in this task is Nepenthes, which can capture malicious software propagating over the network. With copies of malicious samples at hand, they can be analyzed to understand their capabilities. (Coincidentally, I teach a SANS Institute course about this.)

- Understand the intruder's intentions by observing his interactions with the compromised environment. This can be accomplished by deploying a series of honeypots to fool the intruder, whether a human or a program, about the authenticity of the targeted system. The bootable Honeywall disk, distributed for free by the Honeynet Project, can help enable this, and includes excellent monitoring tools.

- Determine whether your users visited malicious websites by employing a client-side honeypot that crawls and examines Web pages. Drive-by downloads, which exploit vulnerabilities through the Web browser, are a common infection technique. Consistently blocking this threat vector may be hard, but you can still detect the incident quickly. If your organization has a mechanism, such as a proxy server, that records visited URLs, you can use the free Caffeine monkey tool from Secure Works to automatically examine those sites for Web exploits.

The most challenging aspect of using honeypots is deploying them in a manner that prevents an intruder from using them as a launching pad for attacks. If your organization chooses to experiment with honeypots, be sure to implement the safeguards outlined in each tool's documentation. For an overview of honeypots and deployment scenarios, see the book Virtual Honeypots by Niels Provos and Thorsten Holz.

3.9.15 PROTECT THE ENDPOINT FROM MALWARE THREATS

Alas, despite information security's best efforts, malicious software may bypass network defenses and reach a system you're trying to protect. Personal computers are particularly vulnerable, because PCs are often used in unpredictable ways and places. Here are the techniques that can help lock down laptops and desktops:

- Employ antimalware tools with behavior-blocking capabilities. Traditional signature-based antivirus techniques are no longer sufficient. Modern security suites from the familiar antivirus vendors can observe local executables for behavior that characterizes malicious software, such as attempting to monitor keystrokes or writing to certain registry locations. This helps detect malware that evades signature detection and block its actions. However, before enabling such tools across the enterprise, be sure to confirm they do not interfere with regular business activities.

- Look out for rootkits. Though far from being a novelty, only recently have rootkits found their way into "mainstream" malware. Rootkits' stealthy capabilities make it particularly difficult to detect an infection. Fortunately, anti-malware products are becoming better about detecting rootkit-concealed malware. They do so mostly by identifying inconsistencies in the way different OS components describe the system's state. Free stand-alone rootkit scanners include GMER, Microsoft Rootkit Reealer, and Sophos Anti-Rootkit.

- Protect browsing activities to anticipate drive-by downloads and other browser exploitation techniques. Hardening the browser may involve creating a protective sandbox around it with a tool such as sand boxie (it's free). It also helps to run the browser with fewer privileges; that's where Vista's built-in User Account Control (UAC) and free tools such as DropMyRights can help. Don't forget to disable unnecessary browser features and components; you can exercise fine-grained control over Internet Explorer with the help of Windows Group Policy.

- Keep up with security patches. Information security pros are getting better at keeping up with security updates for Microsoft products, but knowing when and how to patch third-party software, such as Acrobat Reader and Java Runtime, is more challenging. Free tools that detect missing patches for third-party software include F-Secure Health Check and Secunia Software Inspector.

- Lock down the work station. Last, but not least, is the need to harden the core OS on the endpoint. This involves disabling unnecessary OS components; Group Policy is very helpful for this. It can be used to restrict which applications may run via its Software Restriction Policy feature. A free stand-alone tool that can limit which executables may run is Trust-No-Exe from Beyond Logic.

3.9.16 A COMPREHENSIVE SECURITY PROGRAM IS A MUST

As your organization considers its antimalware strategy, remember that there is no quick fix to this growing threat. Effective approaches incorporate detective and preventative controls that create multiple defensive layers. There are products, both commercial offerings and free tools, to help you along the way. These tools are only as effective as the overall security program that they are a part of.

3.9.17 NATIONAL SECURITY LETTERS

Imagine if the FBI could, with only a piece of paper signed by the special agent in charge of your local FBI office, demand detailed information about your private Internet communications directly from your ISP, webmail service, or other communications provider. Imagine that it could do this:

- without court review or approval
- without you being suspected of a crime
- Without ever having to tell you that it happened.

Further imagine that with this piece of paper, the FBI could see a wide range of private details, including:

- your basic subscriber records, including your true identity and payment information
- your Internet Protocol address and the IP address of every Web server you communicate with
- the identity of anyone using a particular IP address, username, or email address
- the email address or username of everyone you email or IM, or who emails or IMs you
- the time, size in bytes, and duration of each of your communications, and possibly even the web address of every website you visit

Finally, imagine that the FBI could use the same piece of paper to gain access your private credit and financial information — and that your ISP, bank, and any other business from which the FBI gathers your private records is barred by law from notifying you.

Now, stop imagining: the FBI already has this authority, in the form of National Security Letters. These are essentially secret subpoenas that are issued directly by the FBI without any court involvement. Thanks to the USA PATRIOT Act, the only requirement the government must meet to issue an NSL is that the FBI must certify in the letter that the information it is seeking is relevant to an authorized investigation to protect against international terrorism or clandestine intelligence activities.

The number of National Security Letters used each year is classified, but the Washington Post has reported that by late 2005, the government had on average issued 30,000 National Security Letters each year since the PATRIOT Act passed in 2001.

That’s a hundredfold increase over the pre-PATRIOT numbers. Further revelations by the FBI's Inspector General in 2007 showed that in many cases, the FBI had failed even to meet the weak post-PATRIOT National Security Letter standards, illegally issuing so-called "exigent letters" to communications providers asking for the same information National Security Letters are used to obtain, but without meeting the minimal requirement that the requested information be relevant to an authorized terrorism or espionage investigation. EFF has since sued the Department of Justice to learn more about how the government has been abusing its National Security Letter authority.

3.9.18 FOREIGN INTELLIGENCE AND TERRORISM INVESTIGATIONS

All of the government surveillance tactics and standards discussed in previous sections relate to law enforcement investigations — that is, investigations for the purpose of gathering evidence for criminal prosecution. However, the government also engages in surveillance in order to combat foreign threats to national security.

When it comes to foreign spies and terrorists, the government uses essentially the same tools — searches, wiretaps, pen/traps, subpoenas — but operates under much lower legal standards and in much greater secrecy. It's important that you understand these foreign intelligence surveillance authorities such as the government's access to records using National Security Letters and its wiretapping powers under the Foreign Intelligence Surveillance Act (FISA) so that you can evaluate the risk of such surveillance to you or your organization and defend against it.

3.9.19 DEFENSIVE TECHNOLOGY

If you are looking for basic technical information on how to protect the privacy of your data — whether it's on your own computer, on the wire, or in the hands of a third party — Although we hope you'll have the time to review all of the information in the SSD guide, get to the technical details, this is where you can read articles that will explain:

- the basics of the relevant technologies, such as the Internet Basics and Encryption Basics articles

- how to improve the security of different communication applications, such as your web browsers, email systems and IM clients

- how to protect your privacy by using defensive technologies such as secure deletion software, file and disk encryption software, and virtual private networks

- the overarching security threat posed by malware, how to evaluate that threat, and how to reduce it.

Just remember: technology changes quickly. We'll be doing our best to keep these articles updated to reflect current developments, but in the meantime, you should take the time to review information from multiple sources before making any serious security decisions.

3.9.20 INFORMATION STORED BY THIRD PARTIES

Third parties — like your phone company, your Internet service provider, the web sites you visit and interact with or the search engine that you use — regularly collect a great deal of sensitive information about how you use the phone system and the Internet, such as information about who you're calling, who's emailing or I Ming you, what web pages you're reading, what you're searching for online, and more.

In addition to those records being compiled about you, there's also data that you choose to store with third parties, like the voice mails you store with you Cell Phone Company or the emails you store with your email provider. In this, we'll talk about the legal rules that govern when and how law enforcement agents can obtain this kind of information stored by and with third parties. We'll then outline steps that you can take to reduce that risk, by learning how to reduce the amount of information collected about you by third parties, minimize the amount of data you choose to store with third parties, or replace plainly readable data with encrypted versions for storage with third parties.

3.9.21 DATA STORED ON YOUR COMPUTER

Search, Seizure and Subpoenas. In this, you'll learn about how the law protects — or doesn't protect — the data that you store on your own computer, and under what circumstances law enforcement agents can search or seize your computer or use a subpoena to demand that you turn over your data. You'll also learn how to protect yourself in case the government does attempt to search, seize, or subpoena your data, with a focus on learning how to minimize the data that you store and use encryption to protect what you do store.

OBJECTIVES

Here we deals with the Descriptive And Detecting Malware’s main function which is to serve as a standard method of characterizing Malware based on its behaviors, artifacts, and attack patterns. By identifying the distinct attributes present in Malware instances etc.

3.10. NON-SIGNATURE MALWARE DESCRIPTION

Under this broader concept, Descriptive And Detecting Malware’s main function is to serve as a standard method of characterizing malware based on its behaviors, artifacts, and attack patterns. This will allow for the description and identification of malware based on distinct patterns of attributes rather than a single metadata entity (which is the method commonly employed in signature-based detection).

Characterizing malware using a standardized method is not a new concept; however, the primary means of doing so are still rooted in the development of some form of signature (e.g. physical, functional, etc.) and are narrow in scope.

On the other hand, more detailed characterizations that exist tend to rely on narrative text to describe the outputs of dynamic and static analysis. Descriptive And Detecting Malware’s focus on structured attribute-based characterization provides several capabilities that the aforementioned methods do not possess. These capabilities stem from existence as a domain-specific language, with an encompassing and unambiguous vocabulary and grammar. Such a vocabulary and grammar will provide the means for standardized characterization and communication of any malware attributes, and will therefore enable the following capabilities.

3.10.1 MALWARE GROUPING

By identifying the distinct attributes present in malware instances, the similarity of two or more malware samples can be gauged. In this fashion, malware with the same attributes can be grouped together. This will permit accurate association of functionally identical polymorphic/metamorphic malware samples, and therefore help eliminate the duplication of analysis efforts.

Similarly, such characterization enables the description of malware variants based on small differences. This permits the partitioning of malware instances based on shared and unique attributes. Malware families and their variants can therefore be identified by defining the base group of attributes common to a family, as explained in 3.3.

3.10.2 BLENDED MALWARE THREAT DESCRIPTION

An increasing number of malware instances have characteristics that can be ascribed to multiple types of malware [13]. Currently, no common method exists for communicating that these instances share these normally separate characteristics; instead, they are simply typed based on their perceived dominant attributes. However, such blended malware threats can be accurately described on the basis of their constituent attributes, thereby eliminating any confusion as to their true function.

3.10.3 MITIGATION OF ARMORING TECHNIQUES

Attribute and behavioral based malware characterization will also mitigate the challenges posed by polymorphism, metamorphism, encryption, packing, and other obfuscation and armoring techniques. These techniques focus on circumventing specific detection methods through various means; however, the core functionality, or semantics, of a malware instance will always remain intact, and can therefore be characterized based on its attributes.

3.11 UNAMBIGUOUS MALWARE TYPING

In general, classifying an object based on some pre-defined category is often useful for conducting systematic studies. For this reason, malware has been binned into types since its emergence as a serious security threat. However, without a standardized format for accurately identifying the characteristics that define these aforementioned malware types, such typing has been fairly arbitrary and inconsistent. As a result, this has lead to confusion about what exactly a malware type entails, and has subsequently diminished the value of malware typing.

This top-down, tree-structured approach to malware typing is also not appropriate for the description of blended malware threats, as detailed in this book. Having such a fixed tree for typing works only if the malware instance being typed contains attributes that are representative of a single malware type; however, once the malware contains attributes that may belong to two or more malware types, such a tree will fail to accurately classify the malware.

Likewise, such inaccurate and ambiguous typing methods can be problematic in conjunction with heuristic malware detection, where a generic malware type is reported instead of an exact malware instance. Since this reported malware type is information that can be used for threat assessment and response, an incorrectly reported type may lead to an inadequate response and augmented damaging effects. For example, a heuristic engine that incorrectly classifies a network worm as a Trojan based on faulty typing may delay the normal quarantine operation that may be part of an organization's response to network worms and lead to further infections.

Therefore, due to the fact that the various types of malware in the wild have distinct sets of shared attributes, one aim of Descriptive And Detecting Malware is to permit empirical and unambiguous malware typing based on these attributes. To accomplish this, a large sample set of malware would be characterized using Descriptive And Detecting Malware‟s. Afterwards, the malware instances that share a predefined number of certain types of high-level attributes (i.e. mechanisms) would be grouped together; each such group would therefore represent a malware type. While there is likely to be some level of overlap when composing such groupings, giving priority to certain shared attributes for tie-breaks should eliminate most such overlaps.

3.12 IMPROVED MALWARE PEDIGREE TRACKING

As with malware typing, the usage of Descriptive And Detecting Malware in describing malware instances will permit the identification of the minimum set of lower-level attributes (i.e. actions and behaviors) necessary and sufficient for characterizing a particular malware family. This will enable the accurate identification of new instances of existing malware families through the comparison of their Descriptive And Detecting Malware-characterized attributes with the unique set of attributes specific to a malware family. Such identification will also facilitate the identification of code sharing across families. In this manner Descriptive And Detecting Malware should enable accurate and unambiguous malware pedigree tracking.

3.13 LEGAL MALWARE DEFINITIONS

DADM‟s enumeration of malware attributes and their corresponding end-goals will permit the establishment of the behaviors and attributes that have been commonly observed as belonging to malicious software. Along with further analysis of individual malware samples, this will enable the creation of legal definitions of malware.

For example, back-up software that copies files from a user‟s hard disk to an off-site web server is typically classified as benign if it is installed by the user and the aforementioned actions are performed at the behest of the user. A program that performs the same actions can be classified as malicious if it is intended to steal files by being installed and executed transparently, without the authorization and permission of the user.

Thus, in this second case, transparency of insertion and non-user initiated execution are the attributes that differentiate benign versus malicious intent, and these important attributes would be included in the Descriptive And Detecting Malware characterization of the program.

However, the majority of software behaviors can be classified as being either benign or malicious, with the context of their use quite often being the sole differentiator between the two. A program, with the same exact Descriptive And Detecting Malware characterization as the malicious example described above, could also be used for benign reasons by a system administrator to back-up a user‟s system without the user‟s knowledge.

Clearly, Descriptive And Detecting Malware could be used to define the groups of behaviors and attributes that have the potential to be malicious based on past observation. If a behavior has been observed multiple times in malware and has been identified as being consistently used for information theft, chances are that any other program that implements that same behavior is malicious too, as long as it implements the other known malicious behaviors. However, the concrete establishment of the intent of a piece of software based on its attributes and behaviors is a complex problem and one that is outside the scope of Descriptive And Detecting Malware.

CHAPTER 4

SCOPE AND PROPOSED FRAMEWORK

SCOPE

Descriptive And Detective Malware (DADM) is being developed as a language for addressing all known types, variants and manifestations of malware. Although initial work focuses on supporting the characterization of the most commonly discussed types of malware (such as, drive-by-downloader, Trojans, worms, kernel rootkits, etc.), DADM will be applicable to the characterizing of any of the more esoteric malware types. This can range from malware that is currently only a proof-of-concept, such as a hypervisor rootkit, to firmware and malware introduced into a product through its development activities and components, tools and library supply-chain.

Accordingly, DADM is intended to standardize malware characterization and communication, and subsequently provide the wider security community with a useful platform for conducting and integrating their anti-malware operations. DADM is not meant to supply unique identifiers for malware instances, nor subsume the work done by the AV industry in creating malware detection mechanisms and anti-malware countermeasures.

PROPOSED FRAMEWORK

As a language and format for attribute-based malware description, DADM‟s core components include a vocabulary, grammar, and form of standardized output (Figure 11,)

The enumerated vocabulary is composed of three distinct levels of malware attributes, as well as any metadata. DADM‟s schema is effectively a grammar and defines the structure of the enumerated elements and the relationships between them. Finally, the DADM cluster is a standardized format for the output of any DADM characterized data.

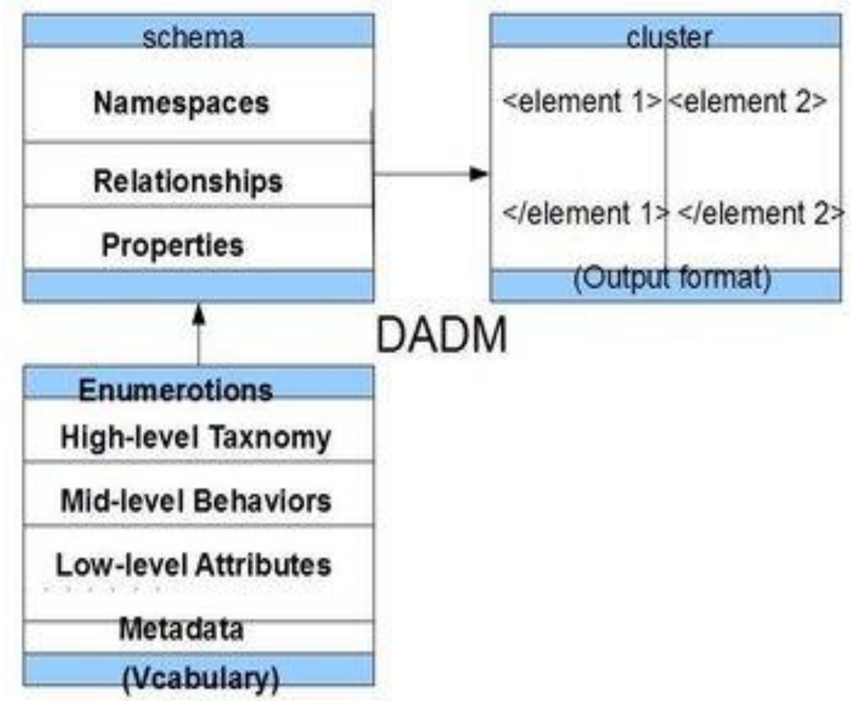


Figure 11: DADM Overview

4.1 DADM’S ENUMERATED ATTRIBUTES

At its heart, DADM will be composed of three tiers of enumerations of malware attributes (Figure 12, below). Each tier will consist of a finite number of attributes and will likely decompose into multiple levels of abstraction for more accurate malware characterization. Since software can only perform the actions that can be achieved through execution of the finite instructions provided by the underlying system’s Instruction Set Architecture (ISA), and because these instructions consequently have a fixed number of arguments, it follows that these attributes are finite and enumerable. Therefore, DADM‟s enumerated attributes will include the detailed actions and behaviors performed by the software, as well as the mechanisms that these actions and behaviors serve to implement.

DADM's ability to communicate high-fidelity information about malware will be directly tied into its ability to accurately characterize the numerous types of malware in existence, as well as those created in the future. Therefore, at a minimum, DADM must be able to describe any low-level actions performed by malware. However, its utility would be significantly degraded if it did not also have the ability to group such information into higher-level representations of malware behavior.

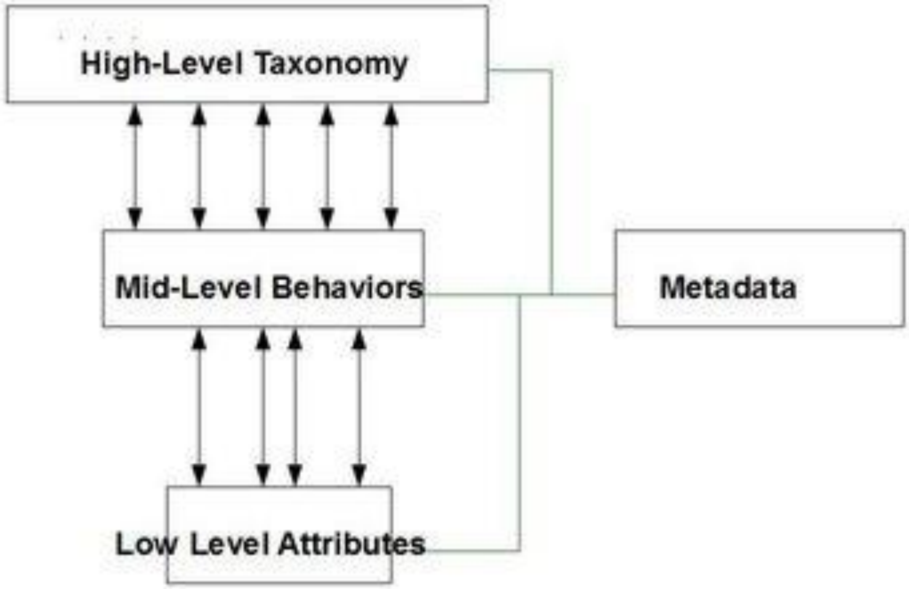


Figure 12: DADM's Tiers of Enumerated Elements

4.1.1 LOW-LEVEL ATTRIBUTES

At its lowest level, DADM will describe attributes tied to the basic functionality and low-level operation of malware. This can include observable entities such as system state changes (e.g. the insertion of a registry key), as well as any features extracted through the disassembly of malicious binaries (e.g. specific assembly instructions). Therefore, likely sources of such data include static analysis, dynamic analysis of malware binaries through sandboxes, network and host-based IDS, and IPS.

The specific level of coverage and its composition at this level is still an open issue. There are certainly DADM use-cases relating to the characterization of assembly code and other features specific to static malware analysis. It would likewise be useful to have the ability to describe the algorithms employed by malware and the nature of their operation. However, it may make the most sense to enumerate and support such attributes at the mid, behavioral level, defined in the next section.

As the community defines DADM, the enumeration and definition of these low-level attributes will need to be done first, as this will likely make it easier to gain consensus on the mid-level behaviors. It would also be useful to leverage the work performed by the IEEE"s ICSG [27] in creating a format for exchanging certain types of low-level malware attributes and metadata.

4.1.2 MID-LEVEL BEHAVIORS

At the middle tier, DADM‟s language will abstract the aforementioned low-level attributes for the purpose of defining mid-level behaviors. The definition of such behaviors gives insight into the consequences of the actions performed by the low-level attributes, and allows a higher-level representation of these actions to be constructed.

For instance, the description of a registry key created or modified by malware is a fairly abstract bit of data that can be useful for detection purposes. However, it does not provide any insight into why the malware created or manipulated the registry entry. Such a registry entry could have many possible uses, including being used to ensure that the malware gets executed at system start-up, or as a simple flag to indicate that the system has been infected. Including the necessary components for characterizing such mid-level behaviors in the DADM language will allow for the accurate description of the intent or goal behind system state changes and other low-level malware attributes.

Since these mid-level behaviors serve to connect the tiers of the low-level attributes and high-level taxonomy, and are the least well-defined of the three (representing consequences rather than distinct attributes or mechanisms), this enumeration will likely be the most difficult to construct and gain consensus on. This development process will likely be iterative and require many passes in order to accurately cement the relationship between these behaviors, the low-level attributes that they abstract, and the high-level taxonomy that directly references them from above.

4.1.3 HIGH-LEVEL TAXONOMY

At the more conceptual and high level, DADM‟s vocabulary will allow for the construction of a taxonomy that abstracts clusters of mid-level malware behaviors based upon the achievement of a higher order classification or grouping. We envision that such taxonomy will have views (i.e. unique layouts) intended for different target audiences – for instance, forensic analysts may only be interested in looking at observable malware payload behaviors, etc.

To expound upon the example given for the mid-level behaviors, ensuring that malware is executed at start-up is a behavior that is typically part of a persistence mechanism. This behavior is often accompanied by the instantiation of a binary copy of the malware on the local machine. Therefore, in DADM‟s top-level taxonomy, these two mid-level behaviors would be defined as belonging to the class of persistence mechanism.

Once DADM‟s mid-level behaviors have been defined, the DADM community will be in a position to begin the process of creating the high-level taxonomy and constructing the appropriate and necessary behavioral linkages. Accordingly, it would be useful to leverage the work done on defining categories of high-level malware mechanisms, such as that performed by SANS‟ Internet Storm Center [24]. Some of these mechanisms we have described using our own definitions (Appendix A.II).

4.1.4 METADATA

In order to include all pertinent information regarding malware and to fully describe the common actions of malware and the rationale behind them, DADM will characterize malware-appropriate metadata.

This can range from metadata associated with malware behaviors, like the transparency of the insertion mechanism used, to the more common types of metadata that are associated with malware artifacts such as file hashes. However, defining the exact nature of malware-related metadata is an open question, and one that is best answered with the involvement of the larger security community.

4.2 THE DADM SCHEMA

DADM's enumerations of behaviors and other attributes are necessary for the establishment of a vocabulary for characterizing malware. Therefore, the primary intent of DADM's schema is to define syntax for this vocabulary. Likewise, the schema serves to create an interchange format for the DADM language, and can also be utilized as a baseline for the creation of malware repositories or intermediate format for the sharing of information between repositories.

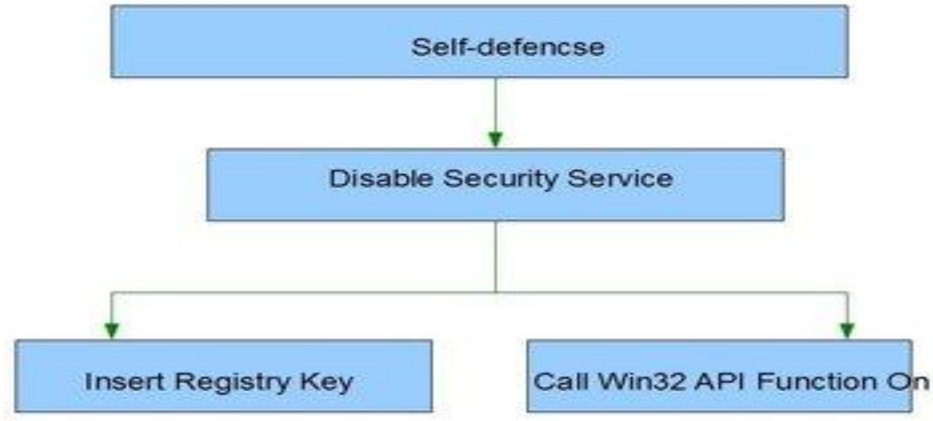


Figure 13: Example of Structure Imparted Through DADM’s Schema

An example of several very simplistic relationships that DADM‟s schema would impart is shown in Figure 3, above. Here, the schema defines (through the „hasMidLevelBehavior‟ relationship) that thehigh-level self-defense mechanism has an associated mid-level behavior of disabling a security service. This behavior, in turn, is defined as being composed of two low-level attributes, namely the insertion of a registry key, and a specific Win32 API function call, through the „composedOfLowLevelAttribute‟ relationship.

The schema will be used to define relationships beyond those dictated in DADM‟s three-layer hierarchy. For instance, a high-level mechanism could make use of multiple laterally associated mid-level behaviors. In this case, the schema will provide the means for correctly defining this lateral relationship. There may be multiple ways of defining such relationships, but only one should be specified in the schema, in order to ensure coherence between DADM characterizations.

Likewise, the size and partitioning of DADM‟s schema is another open issue. It is likely that a single, complete DADM schema will be of substantial size and complexity, thus making it difficult to learn and effectively utilize. As such, this would compromise the adoption of DADM by analysts and researchers due to its substantial learning curve and unwieldiness. Therefore, partitioning the DADM schema into multiple sub-schemas, based on use case, is a possible solution; however, this is another decision that is best made using input from the greater community.

The mechanisms for defining temporal relationships in malware would also be established as part of DADM‟s schema. This would be particularly useful for describing malware that lies dormant before executing its payload, or performs some actions in a repeating sequence, and would permit the accurate identification of where an active malware instance is in terms of its execution flow.
Consequently, at a minimum, the schema will define the following elements.

4.2.1 NAMESPACES

Namespaces in DADM‟s schema will represent the grouping of the behaviors, mechanisms, and other malware-related enumerated attributes of DADM‟s language into well-defined classes. Where applicable, namespaces will follow those utilized by relevant Making Security Measurable (MSM) standards.

4.2.2 PROPERTIES

DADM‟s schema will contain a general set of properties applicable to malware attributes and namespaces, with specific properties for behaviors. This can encompass things like the number of times a behavior occurs, whether it is the child or parent of another behavior, etc.

4.2.3 RELATIONSHIPS

DADM‟s schema will include an established set of rules for defining the potentially multi-directional relationships among namespaces. For example, members of the namespace of mid-level behaviors can be composed of multiple members of the low-level attributes, while members of the low-level namespace can be associated with (but not composed of) only a single member of the mid-level behavior namespace.

4.3 THE DADM CLUSTER

The DADM cluster (Figure 14, below) represents a standard output format of DADM, with the purpose of encompassing any set of attributes obtained from the characterization of a malware instance. Therefore, it will serve as a container and transport mechanism for use in storing and subsequently sharing any DADM-characterized information. A DADM cluster could be used to describe anything from a particular insertion method (composed of several low-level attributes and mid-level behaviors), to all of the attributes of a malware instance, to the key behaviors common to an entire malware family.

Although a DADM cluster will be most useful when encompassing a set of malware attributes with a particular significance (like the insertion method or family behaviors mentioned below), it is intended to serve as a generic container for DADM-characterized malware data. Therefore, it can be used with as little or as much information as desired; any further meaning beyond the explicit data stored in the cluster is defined by its producer. An open issue related to the DADM cluster is how to usefully incorporate large datasets and Detection in DADM. In many cases with malware analysis, there will be large volumes of strings and other static data extracted from malware. Such data is useful from both an analysis and detection standpoint, and would therefore be important for inclusion in a DADM cluster. However, the incorporation of such data along with the DADM characterization in a single cluster could greatly impair readability. Therefore, it may be prudent to store such data in another, external DADM cluster and then reference this secondary cluster inside the cluster containing the main analysis. An auxiliary cluster that uses some other storage method and is referenced by the main DADM cluster is another possibility.

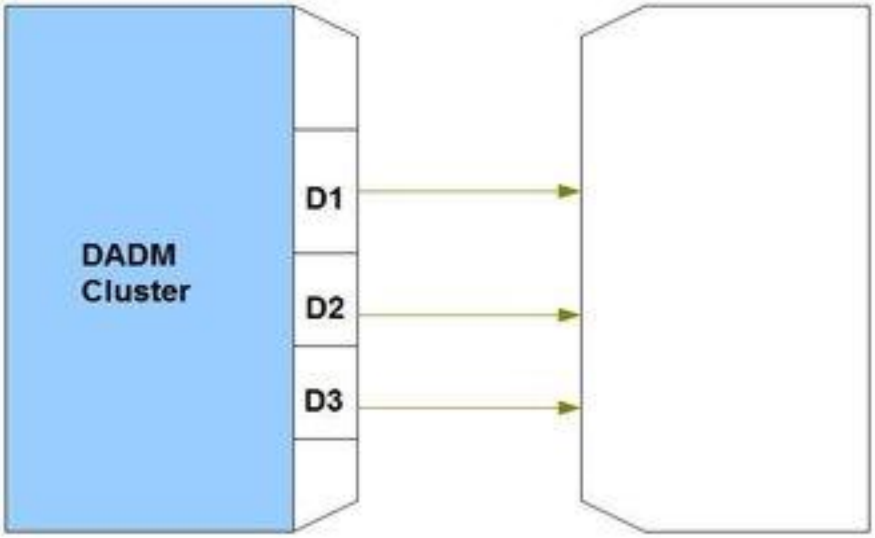
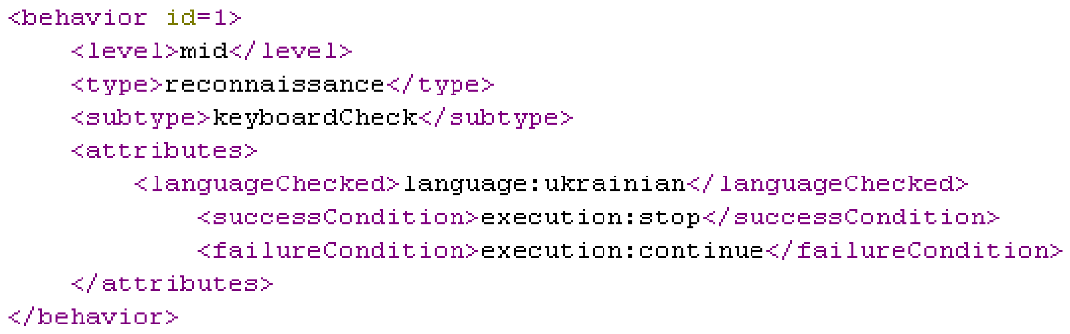


Figure 14: DADM **Cluster Overview**

4.3.1 OUTPUT FORMAT



```
<behavior id=1>
    <level>mid</level>
    <type>reconnaissance</type>
    <subtype>keyboardCheck</subtype>
    <attributes>
        <languageChecked>language:ukrainian</languageChecked>
            <successCondition>execution:stop</successCondition>
            <failureCondition>execution:continue</failureCondition>
    </attributes>
</behavior>
```

Figure 15: Notional DADM **Cluster – XML Representation**

As shown in Figure 15 above (intended only to illustrate a possible form of DADM XML output), DADM will initially support XML as a data interchange format due to its ubiquity and propensity for being human-readable as well as machine-consumable. In the future, it is likely that DADM will support Resource Description Framework (RDF) as well as JavaScript Object Notation (JSON) and other such lightweight formats.

CHAPTER 5

DESCRIPTION TEST CASE

The computer worm best known as “Conficker” first appeared in the wild in November 2008, and has since been responsible for the creation of a sizable botnet [28]. Although this malware family does not implement any particularly novel mechanisms, it represents a highly complex, blended threat that makes use of a variety of attack vectors. There are also multiple variants of Conficker in existence, and its multiple layers of obfuscation and self-defense have made it one of the more difficult malware instances to analyze.

In terms of malware instances, Conficker represents a bit of an anomaly in that it was heavily analyzed and studied by the security and anti-malware communities. As such, there is a large amount of detailed information available about its structure and internal mechanisms. This is largely the reason that we chose Conficker as a description test case for DADM and a sanity check for how its three-tiered framework deals with real malware, as having these rich sources of information regarding the malware attributes and behaviors of a particular malware family would permit us to make a more accurate assessment in this regard.

Unlike Conficker, the vast majority of malware instances are not analyzed to any large degree. Therefore, we recognize that having this amount and type of information about a malware instance or a family is a highly uncommon occurrence. However, our hope was that the description of this unique malware instance would be useful for identifying any questions and/or problems related to our current concept of DADM, while also helping to determine any critical missing pieces..

5.1 INITIAL DESCRIPTION

For our first iteration, we used SRI‟s detailed Conficker analysis [29], and attempted to piece together a full DADM characterization of the Conficker. A variant. Unfortunately, we quickly discovered that extracting such information from prose plain text is a tedious and time-consuming task, and that we would require multiple sources in order to construct a more complete Description. Therefore, as a starting point, we decided to categorize SRI‟s top-level control flow for Conficker.A.

Although this process was fairly straight forward, due to the relative simplicity of the control flow, it brought up an interesting issue. The control flow‟s mixing of low-level attributes and mid-level behaviors made it difficult in some cases to accurately identify the boundaries between the high-level mechanisms being employed. This is because it is hard to judge the purpose of a low-level attribute (for instance “create random name in system32 directory”) without any accompanying information; consequently, it is not possible to link a low-level action with the high-level mechanism it belongs to without knowing its function.

Of course, it is unlikely that every analysis will have a clear link between low-level attributes and mid-level behaviors. Likewise, not all users of DADM will require such information. Therefore, this reiterated the notion that DADM must be flexible enough to characterize malware using any information that happens to be available.

5.2 SECOND DESCRIPTION

For our second description, we stayed closer to our original intent to gain a broader sense of how DADM might be applied. Accordingly, we utilized multiple data sources [30-34] in order to have a diverse set of information to work with that would permit the construction of a more complete characterization.

[2] This time around, we decided to focus on the second variant of Conficker seen in the wild, commonly referred to as “Conficker.B”; this variant includes a number of interesting mechanisms and behaviors not present in Conficker.A, such as self-defense. Our process for this Conficker.B consisted of two steps; first, we binned any attributes into the bottom two DADM tiers (for the sake of simplicity we focused only on observables at the low level – see below for further discussion) Next, we attempted to link the mid-level behaviors as closely as possible with a high-level mechanism from our pre-defined set (Appendix A.II).

Overall, this bottom-up approach was fairly intuitive. By explicitly specifying the action performed by each low-level attribute (e.g., create, execute, modify, etc.), along with the object that it was performed on, we were able to better define and group the mid-level behaviors. Of course, this is based on the assumption that such behavioral information was present in the analysis.

Figure 16, below, shows the result of Conficker.B‟s Windows File Sharing propagation mechanism being characterized in this manner. Here, downward arrows symbolize the logical progression of the mechanism‟s execution, and dotted lines symbolize conditional elements that must be met for the execution to continue. Our complete characterization of Conficker.B done in this fashion can be found in Appendix B at the end of this document.

Consequently, this allowed us to create a very basic grammar for the low-level observable attributes, which we tried to utilize consistently throughout our description. This grammar consists of an action, the type of object that the action is performed on, and the specific object which the malware instance performs the action on. For example, Conficker.B patches Windows‟ netapi32.dll to disable the vulnerability that it used as a vector for successfully inserting itself onto the compromised machine. Using our simple grammar, this action would be written as:

Modify File: \%system32%\netapi32.dll

Although we were largely successful in binning the analysis information we found into the three defined DADM tiers, we had several observations and associated open questions throughout this process, as detailed in the following subsections. A more thorough discussion on a number of these topics stemming from our Conficker characterization, as well as other DADM issues and challenges, can be found in chapter 8.

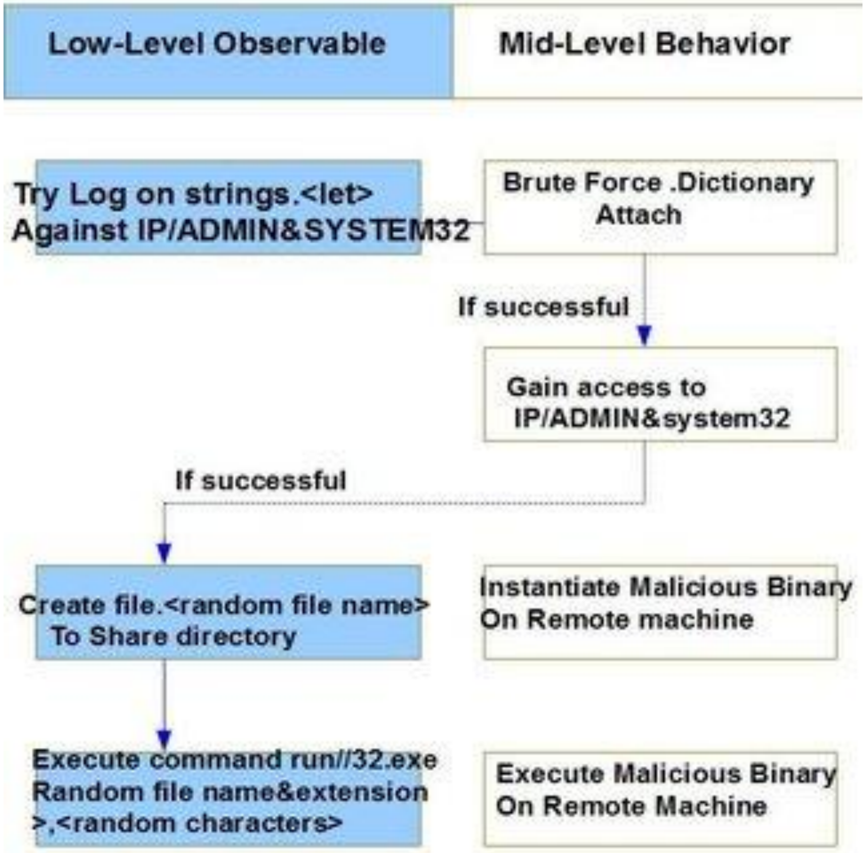


Figure 16: Conficker.B File Sharing Propagation Mechanism

5.2.1 GENERAL OBSERVATIONS

The following are general observations on DADM and its conceptual features that were drawn from our characterization of Conficker.B. Most of these observations relate to the fact that DADM and its elements must ultimately be defined in much greater detail for the construction of accurate malware characterizations to be fully operative.

5.2.1.1 ATTRIBUTES

We first noticed that the level of detail of each of DADM"s tiers must be explicitly defined in order for attributes to be binned uniquely. For instance, in Figure 5 above, "try logon strings" is binned as a low-level attribute; however, this attribute could also have a number of commands/API calls associated with it, thereby actually making it a mid-level behavior (i.e. the purpose behind the commands/API calls).

In this case, such information was not available in the analyses that we utilized, so we decided to make this a low-level attribute associated with a brute-force attack behavior; such cases of ambiguity must be properly addressed and resolved during the continued development of DADM.

We also discovered that low-level attributes can encompass more than pure observables, as they can refer to changes of state, transient behaviors, and other features discovered through analysis. While we had originally taken this into account, our initial concept of DADM made mention only of observables, for the sake of simplicity. Now that we had real-world data to analyze, it was clear that we needed to explicitly state which entities other than observables that low-level attributes can refer to.

Finally, it was readily apparent that we needed to overtly define the hierarchy within each of DADM‟s three tiers, or else risk ambiguity in such characterizations. This is particularly true for the high-level tier, where we had questions regarding the accurate positioning of the mechanisms that we were binning the mid-level behaviors into. For instance, when referring to a specific propagation mechanism, such as the one that Conficker.B uses in conjunction with Windows File Sharing, it is clear that this mechanism belongs hierarchically as a branch off the general propagation mechanism. Similarly, armoring and obfuscation may lie under the umbrella of the self-defense mechanism.

5.2.1.2 SCHEMA

With regards to the schema, we discovered that the analyses we used for our sources could effectively be partitioned into two categories; those that center on thorough analysis through reverse engineering, and those that have less detailed analysis and are more focused on detection. By having the multiple DADM tiers (even if they‟re hard to populate), we are providing a framework for connecting these points of view. Thus, DADM has the potential to ensure information flow between these two processes, which while being inter-connected, are seldom integrated.

5.2.2 OPEN QUESTIONS

The next few subsections describe several open questions that were the result of our characterization of Conficker.B. Many of these questions are associated with the specific implementation of DADM and are discussed further in section 8.1.

5.2.2.1 ATTRIBUTES

After we noticed that low-level malware attributes can consist of entities other than observables, we also discovered that many analyses, especially those that go into considerable depth, characterize some form of machine code. Such code can provide valuable insight on the internal operation of malware, particularly with regards to obfuscation/encryption mechanisms and the like; it is therefore a useful attribute that has as much utility as other DADM attributes. However, how to characterize such code is an open question, especially considering that it can have multiple forms, be of substantial length, and is not enumerable.

Likewise, we found that several Conficker analyses made note of “side-effects,” or the consequences of unintentional interactions between the malware and its execution environment.

Such information can be considered a separate observable attribute, and would clearly be useful for detection, but the question of how to include it has yet to be answered, due to its environmental (i.e. platform) dependencies and the general difficulty in accurately correlating such information with malware behavior. One possible method would be to separate observables into categories of “first-order” and “second-order,” with the first-order stemming directly from the execution of the malware, and the second-order being side-effects of execution in a specific environment.

5.2.2.2 SCHEMA

After our characterization of Conficker.B, it was clear that DADM‟s schema would need some way of defining logical and other types of relationships that can exist on their own or between multiple malware attributes. A simple type that we included in our characterization is the conditional relationship (see Figure 5 in Section 6.2 for an example). However, how to include these relationships without adding a significant layer of complexity to the schema has yet to be determined and is an open question.

CHAPTER 6

TIES TO MSM STANDARDS

As part of MITRE's Making Security Measurable (MSM) effort2, DADM will make use of other relevant MSM standards, where appropriate (Figure 17, below).

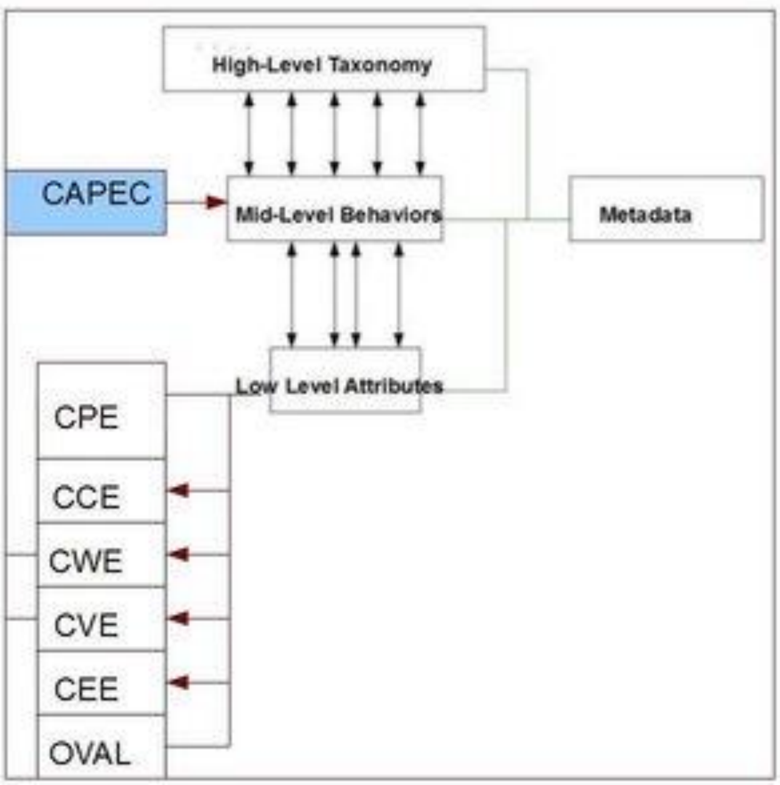


Figure 17: DADM's **Relationship with MSM Standards**

A number of these standards can be utilized for more accurately characterizing malware, especially with regards to standardized reporting and assessment of threats and vulnerabilities in the operational environment. However, due to potential implementation complexities and the varying levels of maturity of these standards, it is likely that only a subset will initially be referenced by DADM.

Likewise, there exists an overlap in the namespace of observable data relating to malware, attack patterns (CAPEC), and logged events (CEE), that must addressed for these three standards be inter-operable.

6.1 CAPEC

DADM will make use of the Common Attack Pattern Enumeration and Classification (CAPEC) 3 for describing the relevant attack patterns associated with the high-level malware taxonomy, such as those dealing with network reconnaissance, propagation, insertion, and command & control. DADM"s usage of CAPEC will allow for such behaviors to be defined through an industry standard attack pattern enumeration, thus ensuring that the attacker's perspective in implementing these behaviors is properly represented.

This association will also provide researchers with detailed information regarding the behavior's motivation (if included in the CAPEC entry). It is conceivable that such information could be utilized by researchers for determining the over-arching intentions of the malware author (by abstracting multiple CAPECs and other malware behaviors), as well as by developers for creating software with improved security against malware.

6.2 CEE

DADM will use the event description language provided by Common Event Expression (CEE) 4 to describe logged events associated with malware activity. Such entries can be linked to specific malware behaviors and used to determine the presence of malware.

6.3 CWE

If it is determined that a malware instance exploits a particular software weakness, DADM will link to its corresponding Common Weakness Enumeration (CWE) 5 entry. This linkage will allow for the generation of statistics with regard to the most common types of weaknesses being exploited by malware, thereby highlighting the areas where better security-oriented coding practices need to be implemented. It will also provide an attribute for use in characterization and correlation when a specific CVE or CCE isn’t is targeted by malware.

6.4 CVE

DADM will link to the Common Vulnerabilities and Exposures (CVE) 6 entry associated with a particular vulnerability exploited by malware. This will allow users to determine the nature of the vulnerability being exploited by the malware, as well as for automated fix and patch assessment through CVE-compatible tools. Likewise, DADM's link to CVE will substantiate vulnerability-based threats by providing a concrete example of their exploitation, which will permit the prioritization of software vulnerability patching and associated threat assessment efforts.

6.5 CPE

For a standardized description of the software and hardware platforms targeted by malware, DADM will make use of the respective Common Platform Enumeration (CPE)7 entry associated with the platforms, permitting the tool-based identification of potential victim machines by IT administrators. Linking to CPE will also allow for assessment of the threat that a malware instance poses to organizational computing resources based on the platforms that it targets.

6.6 CCE

The linkage of DADM to the Common Configuration Enumeration (CCE)8 will allow for the description of any of the vulnerabilities associated with malware that are not related to software flaws. This will allow for the host-based detection of specific configuration-related vulnerabilities exploited by malware, as well as the detection of general configuration issues that malware could potentially exploit. DADM's link to CCE can also substantiate non-flaw based vulnerability threats by providing a concrete example of their exploitation, which will permit the prioritization of configuration vulnerability mitigation strategies and associated threat assessment efforts.

6.7 OVAL

Certain low-level malware attributes may represent attempts at software vulnerability exploitation, meaning that such entries can be linked to corresponding Open Vulnerability Assessment Language (OVAL)9 definitions (if in existence). Such a connection would allow for improved malware threat mitigation, by tying in the ability to easily check for the host-based existence of a vulnerability that is directly associated with a particular malware instance.

Likewise, it can narrow down the potential malware variants capable of infecting a system by correlating the un-patched vulnerabilities present on a system with those linked to by DADM characterizations. Similarly, it could enhance remediation by providing an automated means of checking for any remaining malware artifacts.

OVAL can also be used to determine malware presence based on comparison of multiple scans. A common malware behavior is to patch the particular vulnerability used to exploit a system after successful infection, so that detection of such a “silently” patched vulnerability can be used to establish the presence of malware.

CHAPTER 7

USE CASES

As a domain-specific language for the characterization of malware, DADM serves to provide a vocabulary and grammar for the encoding and decoding of information. It follows that the majority of the use cases for DADM are motivated by the unambiguous and accurate communication of malware attributes that it enables.

While there are a number of ways that DADM-encoded information can be utilized in some automated form, the majority of DADM's use cases are human-oriented. That is, while DADM will provide a foundational basis and structure for use in characterizing malware, we believe that such characterizations will be interpreted and utilized by humans more often than by machines.

7.1 12 ANALYSIS USE CASES

As shown below in Figure 18 DADM will typically be utilized to encode the data garnered from malware analysis. In such a scenario, malware would serve as an input to whatever method of analysis is being utilized. The results of the analysis would then be classified using DADM‟s enumerations and schema, and a DADM cluster would be generated for use in transportation and communication of the new data.

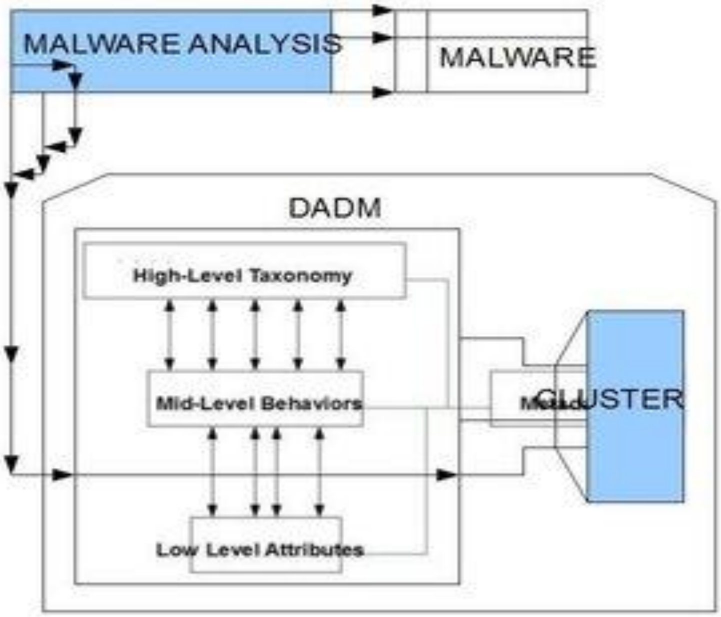


Figure 18: Typical Analysis DADM Usage Scenario

The analysis of malware using static and dynamic/behavioral methods is becoming increasingly important for the purpose of understanding the inner workings of malware. Such information can be utilized for malware detection, mitigation, the development of countermeasures, and further analysis. However, the lack of a common vocabulary for analysis makes it difficult to compare and utilize the results of analyses performed by different analysts and tools.

The encompassing attribute enumerations provided by DADM, containing all possible malware attributes capable of being characterized through malware analysis, will enable the convergence of malware analysis results upon a common vocabulary. Utilization of such a vocabulary for malware analysis should eliminate the confusion and ambiguity resulting from the use of multiple disparate vocabularies for analysis results.

Likewise, through its high-level taxonomy, DADM will provide a way of guiding and helping the analysis process. By including an enumeration of the mechanisms of malware behavior, DADM will give analysts the ability to search for any behaviors and low-level attributes that correspond to these mechanisms, as well as to easily abstract previously discovered attributes and behaviors in multiple dimensions.

DADM clusters could also be utilized as a standard format for use in the creation of visualizations of malware behavior [4]. Such visualizations would permit clear assessment of the low-level actions and mid-level behaviors performed by malware and facilitate natural comparison between two or more malware instances.

7.1.1 BEHAVIORAL/DYNAMIC ANALYSIS TOOL WRAPPER

Current behavioral analysis tools provide a powerful way of characterizing malware activity through its observed behavior. However, there is often no commonality between the output of such tools, thus making comparison and analysis of data between divergent sandboxes difficult. Likewise, re-writing such tools so that their output coincides with a standard like DADM would be unfeasible.

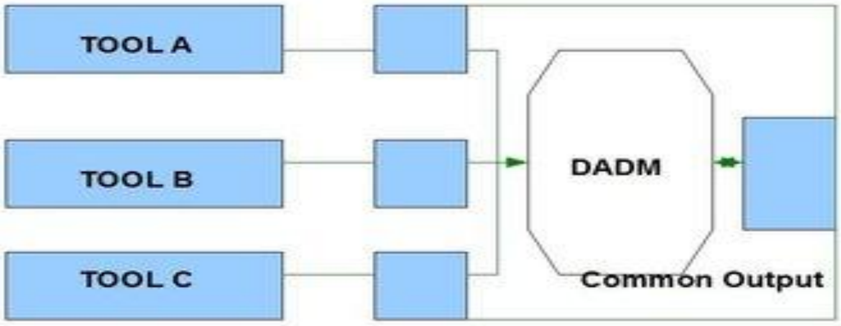


Figure 19: Multiple Tool Output to Common Format

Therefore, for such tools DADM could be utilized to develop a wrapper for standardizing the output of any tool in existence (Figure 8, above). This would ensure data compatibility between divergent tools and facilitate the sharing of behavioral malware data.

Example Scenario

Say that a new sample, representing an unknown instance of malware, is submitted to a sandbox for behavioral analysis. The sandbox reports the analysis results in its own format, but also supports the use of DADM for standardized output. Using the output of this DADM-based wrapper, an analyst is able to compare the data generated with characterizations of previous malware instances performed on different sandboxes. This permits the analyst to establish that this is actually a slightly modified variant of a previously observed malware family.

7.1.2 MALWARE REPOSITORIES

Malware repositories oriented towards analysis often have very specific needs that require the use of a highly customized schema. This entails that sharing or exporting data from a repository defined by such a schema would be very difficult without the existence of a standardized system for the conversion of this data into a common format.

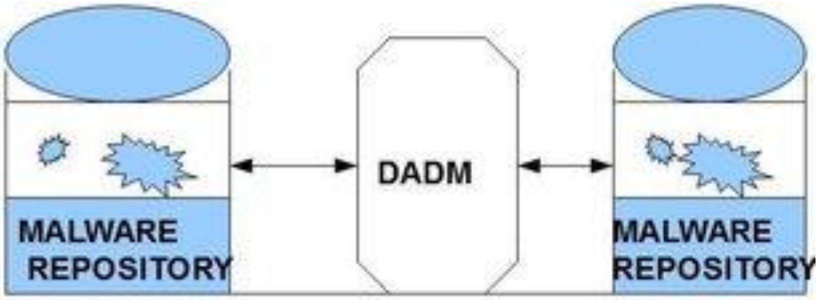


Figure 20: Repository Data Sharing Through DADM

DADM‟s schema could be used for mapping between the dissimilar schemas utilized in malware repositories. This would facilitate the sharing of analysis information stored in disparate repositories (Figure 20, above). Likewise, the usage of DADM in malware repositories would permit improved data-mining due its structuring and labeling of malware attributes.

7.1.2.1 IMPROVED DATA-MINING

Most current malware repository schemas are typically structured with little regard to the information contained inside malware analysis reports, with the entire report often being contained in a single text field inside the database. As such, it is usually possible only to do a string or regular expression based search for information that is highly specific to a single malware instance, such as the name of a file that it drops upon insertion. However, even such relatively simplistic queries are not possible across multiple repositories due to their divergent schemas.

This makes it very difficult to make any meaningful comparison between two malware instances, as there is only one field with a large number of varied and non-structured information to use in the comparison. As a repository grows and expands, this problem becomes even more profound, since the signal-to-noise ratio of simple text-based queries inevitably drops with a larger number of malware samples and their corresponding analysis information in the database.

By mapping a malware repository schema directly to DADM's schema, any ambiguity between malware features and corresponding search queries could be eliminated. Since DADM incorporates low-level attributes, as well as mid and high-level behaviors, the integration of a malware repository schema with DADM would allow for querying the repository based on multiple tiers of discrete malware attributes, and would provide for individual attribute-level comparisons between multiple malware instances.

The aforementioned comparison of discrete malware attributes through DADM could also be used as the basis for a malware similarity metric. In this manner, DADM clusters specified for multiple malware instances could be compared, and a similarity score calculated based on the number of shared attributes at each of the three levels of abstraction.

Finally, organizations may not wish to directly export or share their malware repositories, but could instead allow the execution of search queries against them. DADM could be used in this case to map between repository-independent queries and specific repositories. This would permit the execution of a single query against multiple repositories whose schemas may have little in common. The support of such federated queries is another important aspect of malware data-mining that DADM would enable.
Example Scenario

For example, imagine searching a large unstructured repository for all malware that propagates via spammed infected email messages by using the keyword text of "SMTP." The results from such a query would include some relevant data relating to malware that does in fact use such a propagation mechanism, but could also have information about malware that connects to SMTP servers, attacks SMTP servers, or was originally received in an email message, just to name a few.

If the schema of the aforementioned repository were to be mapped to DADM, querying the repository using the DADM-defined category of (for example) "Propagation Vector: Email/SMTP" would retrieve the desired information, with completely accurate results. While this is a very simple query, it demonstrates the significant data mining capabilities provided by the usage of a domain-specific formal language.

7.1.3 OBJECTIVE CRITERIA FOR TOOL ASSESSMENTS

DADM‟s typing of malware based on discernable attributes can be utilized as objective criteria for use in the assessment of anti-malware tools. In this sense, a tool would be assessed on the basis of its support in detecting all of the attributes associated with a particular malware type. A tool that cannot detect certain DADM-defined attributes associated with a particular malware type can miss any malware that contain such attributes, and therefore cannot objectively be defined as capable of detecting that type of malware.

7.2 OPERATIONAL USE CASES

In the operational domain, threat analysis, intrusion detection, and incident management are processes that deal with all manners of cyber threats. DADM, through its uniform encoding of malware attributes, will provide a standardized format for the incorporation of actionable information regarding malware in these processes. Accordingly, the successful integration of such a standard in the operational environment (Figure 21, below) will facilitate more accurate and enhanced malware triage, detection, and response.

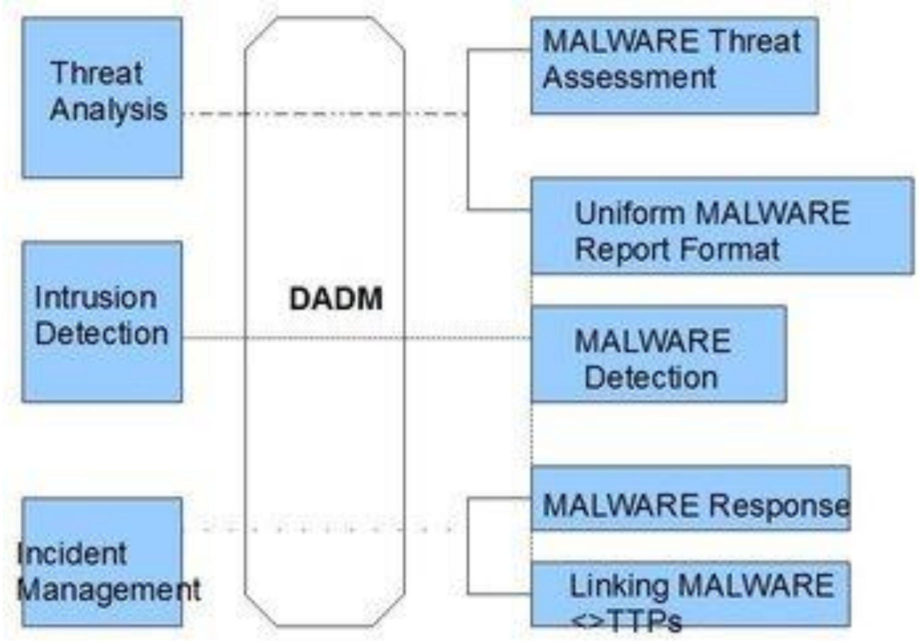


Figure 21: DADM for use in Operations Security Management

7.2.1 UNIFORM MALWARE REPORTING FORMAT

Current malware reporting, while useful for determining the general type and nature of a malware instance, is inherently ambiguous due to the lack of a common structure and vocabulary. Likewise, it often excludes key malware attributes that may be useful for mitigation and detection purposes, such as the specific vulnerability being exploited. Clearly, the current value of malware reporting to end-users is significantly degraded without an encompassing, common format.

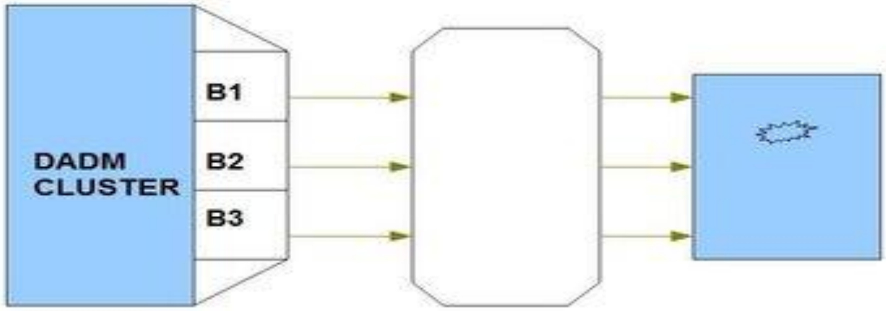


Figure 22: DADM **as a Uniform Malware Reporting Format**

The use of DADM‟s standardized vocabulary and grammar in malware reporting will facilitate the creation of a separate, uniform reporting format (Figure 22, above). Such a format will reduce confusion as to the nature of malware threats through the accurate and unambiguous communication of malware attributes, while also ensuring uniformity between reports composed by disparate authors and organizations.

Example Scenario

For example, say that a consumer of an AV product scans one of their machines and finds that a new piece of malware has been detected. Wishing to know more about the malware, they navigate to the site of their AV vendor and lookup the information regarding the instance that was found. Unfortunately, the information there is formatted as unstructured prose text and is missing key attributes. Likewise, it does not have any information about the vulnerability exploited by the malware which allowed it to get on the system in the first place.

With DADM, such reporting would be structured in a meaningful way. This would permit the consumer in the above example to easily compare the behaviors of this detected instance with those of another. More importantly, it would allow them to quickly identify the vulnerability exploited, by linking to CVE, as well as the specifics of how the malware got on their system in the first place, so they can prevent this infection from re-occurring.

7.2.1.1 MULTIPLE SOURCE INTEGRATION

The majority of useful data in malware reports is currently made up of unstructured narrative text, making it very difficult to accurately compare or combine multiple reports. Therefore, the standard vocabulary and structure of the uniform malware reporting format enabled by DADM will allow for accurate comparisons between multiple malware reports, without the need for converting each report to some intermediate representation.

As such, the use of DADM in malware reporting will permit category-by-category comparison. This will not only allow for identification of differences and similarities among reports, but will also provide for the discernment of critical malware-related areas that need to be focused on or researched further.

7.2.2 MALWARE DETECTION

Characterizing malware based on its attributes with DADM will permit the use of actionable information for malware detection. Specifically, DADM‟s accurate identification of the observable low-level attributes associated with a malware instance will allow for the detection of malware at both host and infrastructure levels.

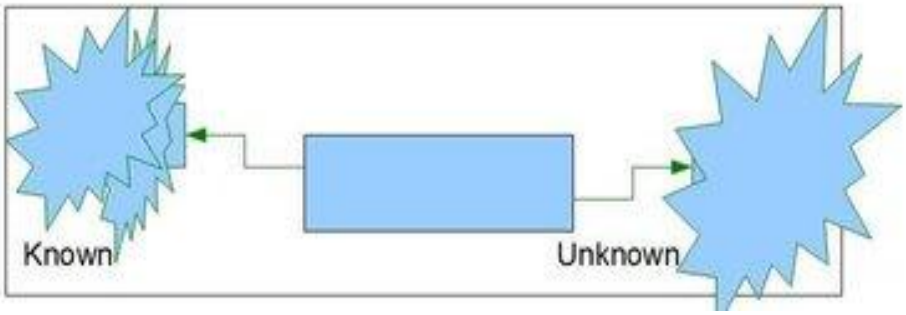


Figure 23: Malware Detection via Shared Attributes

As such, a single DADM characterization of a malware instance, represented by a DADM cluster, can provide data that can be used to detect multiple malware instances. Since there are only a finite number of ways of accomplishing a single software behavior (for instance, malware insertion), particularly at the assembly level, it is statistically likely that there will be an intersection of such attributes between multiple malware instances. Therefore, the DADM characterization of a single instance can permit the detection of malware families and even otherwise un-related malware that have certain attributes in common with the instance (Figure 23, above).

7.2.2.1 HOST-BASED DETECTION

DADM‟s encoding of information regarding low-level system observables will allow for host-based malware detection. This can be accomplished through manual system inspection for DADM-defined entities such as active processes that signal the presence of specific malware instances.
Likewise, utilization of such information contained in DADM clusters will permit tool-based detection. For example, a tool could read a DADM cluster and use the data contained inside to automatically generate OVAL queries for the purpose of detecting the existence of entities such as registry keys and files.

Similarly, using DADM for characterizing malware will permit the definition of patterns of low-level attributes and mid-level behaviors utilized by malware. Such information could be used to establish new rule sets for heuristic-based detection, based on the observed sequences of behaviors.

Example Scenario

For example, say that a new, particularly nasty, malware instance has been analyzed and had a DADM cluster created for it. This cluster details the artifacts and low-level observables specific to the instance, and also gives a behavioral overview of what the malware attempts to do. Accordingly, no signature has yet been created for this malware.

Based on this behavioral overview, the security administrator of an organization determines that this malware has the potential to do serious harm to their computing environment, and that detection of it is a priority. The administrator then takes the DADM cluster for the malware, and uses his DADM-compatible tools to generate from it a number of OVAL queries for use in detecting the artifacts associated with the malware. These queries are then executed across the organization‟s computing infrastructure and the results used to assess whether or not this malware has been detected.

7.2.2.2 NETWORK-BASED DETECTION

Although the use of malware-specific signatures in network intrusion detection and prevention systems (IDS/IPS) has become fairly widespread, there is still no general method for characterizing the relationship between distinct malware attributes and network infrastructure. Such a systematic way of linking malware behavior with outgoing network data could be used for infrastructure-based detection, as well as the accurate identification of observed malware behaviors. Likewise, the accurate characterization of incoming traffic as being associated with a malicious behavior can be used to detect and drop packets associated with behaviors like those employed by bots for command and control.

Therefore, DADM's role as a standard language and format requires that it will be able to characterize any such applicable information regarding malware. In particular, DADM will provide the capability for integrating and characterizing any attributes and patterns related to malicious network activity, thereby permitting network-based malware detection in combination with this data and IDS/IPS. In this regard, we envision the creation of DADM enabled tools that could automatically generate signatures for specific malware behaviors.

Example Scenario

For example, say that a new malware instance has recently been introduced into the wild, and that it exploits unknown network vulnerability as a means of insertion. However, several DADM clusters of this malware have been prepared and integrated in a report describing the new instance. This report characterizes the specific types of network traffic generated by this malware in order to infect systems, as well as its other behavioral and low-level observable attributes.

Using the DADM-encoded information contained in the report, an organization updates their IDS rule sets to block any traffic that looks similar to that of the malware insertion process. Likewise, this information is utilized to detect any machines that have already been infected by the instance. When the exploit is identified and the DADM clusters for the malware instance updated to reflect this, the organization utilizes this information to patch all of their systems, thus making them immune to the insertion of the malware identified by the DADM cluster.

7.2.3 MALWARE THREAT ASSESSMENT

For IT administrators and others charged with protecting systems from cyber threats, one of the most useful aspects of malware reporting is data which details the specific threat that the malware represents. In particular, they are interested in details regarding the specific platforms, vulnerabilities, and weaknesses targeted by the malware. Such data, combined with information regarding the actions performed by the malware, would enable prioritization of malware assessment and change management efforts.

Although current malware reporting may include these useful characteristics, such information has no commonality between reports and does not link to other relevant standards. Therefore, this makes it difficult to ascertain the true threat that malware represents.

DADM"s linkage to OVAL, CPE, CVE, and CWE, will provide system administrators with the necessary information for determining the specific vulnerabilities and weaknesses targeted by malware. Accordingly, DADM"s encoding of mid-level behaviors and a high-level taxonomy will allow for the accurate discernment of the threat that it represents to their organization and infrastructure.

This linkage could also allow for the creation of a malware threat scoring system, similar to that of the Common Vulnerability Scoring System (CVSS) for software vulnerabilities10. As described above, DADM"s link to the relevant MSM standards as well as its characterization of mid and high-level malware features would provide the necessary data for accurately describing the attack vectors and payload of a malware instance. This data could then be used to score the potential impact of the malware based on pre-defined categories, such as payload type (e.g. data theft, bot-like behavior, etc.), propagation, and degree of entrenchment.

Of course, ascertaining such attributes requires a fair amount of analysis, something which is not done for the vast majority of malware samples. Therefore, it is likely that very few instances will link to MSM standards and include the necessary mid-level and high-level attributes. However, there are also patterns relating to malware behaviors and types that can be observed at the lowest level of attributes;

Defining such patterns through DADM will allow an automated, cursory judgment of malware threats to be created using dynamic analysis engines and related methods.

Example Scenario

For example, say that a new malware instance has been seen in the wild, analyzed, and has had a report issued for it. The report uses DADM for encoding the key attributes of interest, and therefore links to the CVE entry detailing the vulnerability exploited by the malware for insertion as well as the CPE entry for the software platform that it targets.

And IT administrator looks at the report, and determines based on its CPE linkage that several of their machines are running the platform targeted by the malware. However, these machines have already been patched against the vulnerability defined in the CVE entry linked in the report. Based on this information, the administrator is able to determine that this malware poses a low threat and detection of it based on the other DADM-encoded information contained in the report has a low priority.

7.2.4 MALWARE RESPONSE

Malware detection alone is often not enough to ensure security and operability. Formulating malware responds, although still a secondary function has become increasingly important.

7.2.4.1 REMEDIATION

One of the current realities with cyber security is that malware detection and prevention of infection is not always a possibility, especially with new and targeted malware threats. Unfortunately, most conventional AV tools and utilities are not capable of removing every trace of a detected malware instance [3]. Thus, even if the explicitly malicious portions of an infection are cleaned from a system (which is not always the case), the remaining pieces may lead to false positives in future scans, thereby potentially leading to a misallocation of remediation resources. Likewise, an incomplete remediation could render a system unstable and prone to future infection.

By providing the means for communicating the exact artifacts and low-level attributes associated with a malware instance, DADM will permit greatly improved remediation of malware infections. Administrators could perform manual remediation based on the data contained in a DADM cluster, or ascertain the remediation performed by another tool by checking for the existence of the aforementioned artifacts.

Accordingly, the incorporation of the temporal element in DADM could prove to be extremely useful for remediation. In this sense, it would provide administrators with an accurate determination of how far along a malware instance is in its execution, and thus allow them to judge the steps that may or may not be necessary for remediation. This could be particularly useful for the remediation of malware that downloads new components or lies dormant for some period of time before executing its payload.

7.2.4.2 COUNTERMEASURES

The development of countermeasures against malware is an ongoing process that must be able to adapt to new techniques employed by malware authors. By correlating specific malware attributes and behaviors with specific counter-measures, DADM will allow organizations to alter their defensive posture based on this link. Accordingly, new malware threats can be ascertained and counter-measures against them developed based on existing links between attributes and previously-developed countermeasures.

Example Scenario

For instance, say that an organization is concerned mostly with the security of their network; any unwanted traffic should not be able to get in or out. Since their security mechanisms and policies revolve around this consideration, the development of malware countermeasures is a high priority, and DADM is therefore utilized to assist in this manner. This is accomplished by analyzing the DADM characterizations of the various types of malware in the wild; this information is then used to block ports and services that are determined to be commonly employed by malware. Accordingly, it permits the development of countermeasures against specific malware mechanisms that could be particularly harmful, such as propagation and command & control.

7.2.5 LINKING MALWARE TTPS

In the analysis of cyber incidents and attacks, it is often meaningful to characterize the tools, techniques, and procedures (TTPs) used in the attack as being part of a set belonging to a particular attacker. When correlated across multiple attacks, such a connection can be useful for the purposes of attribution.

With malware being one of the most prevalent tools employed by attackers, it would be useful to characterize specific malware instances as belonging to a set of tools used by specific attackers. DADM would provide this function, as its standard vocabulary and grammar will permit the accurate identification of the malware attributes observed in previous attacks, and with the “attacker” being defined as a metadata attribute, this would allow for the construction of an accurate link based on previously observed and descriptive malware.

CHAPTER 8

CONCLUSIONS

In this Chapter, the research objectives of this Thesis are revisited and the results of the research based on theoretical and empirical validation techniques in DADM"s are presented before drawing our conclusion and making recommendation for representing some of the high-level challenges related to the development of any complex formal language. This chapter details the issues and challenges relating to DADM that we have identified during its conceptual stage of development. The majority of these issues is with regards to implementation, and will therefore need to be properly researched and addressed. We welcome the community’s input and involvement in resolving these and other issues relevant to the development and usage of DADM. More information regarding participation, including how to sign up for DADM‟s open discussion list, can be found on DADM‟s website11.

8.1 IMPLEMENTATION ISSUES

Due to DADM‟s wide range of use cases, potentially large enumerations, and various attribute properties and relationships, the development process of the language has the potential to be exceedingly complex. Therefore, this process must be carefully planned in order to be successful. It may be necessary to initially focus on a specific subset of use cases, in order to reduce developmental complexity to a manageable level.

8.1.1 DADM SCHEMA STRUCTURING

As DADM has a broad set of use cases, it is clear that most parties making use of DADM will be interested in utilizing only a subset of the language and its features. Therefore, instead of a single monolithic schema, it may make sense to have multiple schemas, based on their applicable use cases. This would potentially make DADM useful to more parties, while also increasing the difficulty of development and consumption of DADM-encoded information. As such, there are trade-offs in terms of complexity, usability, and other issues that must be considered before making any final decisions with regards to the structuring of DADM‟s schema.

8.1.2 TEMPORAL/CHRONOLOGICAL CHARACTERIZATION

Like any piece of software, malware has an execution flow that is a function of its machine code and the environment in which it is executed. Since DADM defines individual attributes relating to specific portions of execution, it needs to have some way of defining any temporal and order-based relationships between such attributes, as discussed in Section 8.2.2.1. This would permit the characterization of attributes based on absolute time, frequency, and order of execution.

Such characterization would especially be valuable for analysis and remediation (as briefly discussed in Section 8.2.4.1), but could also be used for detection. For example, the frequency of a botnet command and control command, if defined in a DADM cluster, could be correlated with network traffic and used as a means of malware detection.

8.1.3 STATIC DATASET INCORPORATION

As briefly discussed in Section 5.3, the integration of large static datasets in DADM is an open issue. Although this could be achieved by including this information in „child‟ DADM clusters which would then be referenced by the „parent‟ cluster, this introduces an extra layer of complexity that may not be worthwhile considering the relatively small nature of the problem. Another option would be to simply make this information optional for display in DADM-based reporting, while still including it inside the DADM cluster used for generating the report.

8.1.4 ASSEMBLY CODE CHARACTERIZATION

Particularly with regards to static binary analysis/reverse-engineering, it would be useful for DADM to have the ability to characterize assembly code, as touched upon in Sections 5.1.1. This would permit the identification of important algorithms, functions, and other non-observable entities that have been previously seen in malicious binaries, while also providing analysts with the ability to describe such code using a standard method. While existing at DADM‟s lowest level, such code is not enumerable, but should still be able to be included in DADM clusters.

Therefore, this issue has two aspects; namely, what format(s) can the code be in for inclusion, and the bigger question of how to characterize such code in a standardized fashion. As far as characterization, one option is to simply tie each block of interesting code to an enumerated behavior from DADM‟s mid-level. However, there may be cases where there are notable instructions existing inside a code block that are worth characterizing separately. Such instructions must therefore also have some method of characterization; it is unlikely that their functionality could be enumerated, so this may end up taking the form of a custom string created by the analyst.

8.1.5 MALWARE METADATA

As briefly discussed in Sections 5.1.4, the specific definition of what constitutes malware metadata and how it relates to DADM is something that must be defined. There are well-established types of metadata relating to malware that will certainly be included in DADM, such as hashes, file lengths, and dates.

However, attributes not representative of core malware functionality, such as packing and encryption mechanisms, also have the potential to be considered as metadata. Therefore, this boundary must be determined in order to eliminate ambiguity in DADM and create a well-defined set of enumerations.

8.2 GENERAL CHALLENGES

The following issues are not directly related to DADM, but rather represent some of the high-level challenges related to the development of any complex formal language.

8.2.1 REPEATABILITY

Repeatability is a potential issue for any language or system intended to characterize such a broad, chaotic landscape as that of malware. The sheer number of potential attributes, combined with the convolution of most modern malware instances, and the different analysis techniques employed, entail that correlating descriptions of malware created using DADM could be problematic. There is also the fact that malware can exhibit “emergent” behavior, unintended by its author(s), through some unforeseen ecological interaction [5].

In theory, a standard such as DADM, if created with a well-defined and unambiguous vocabulary and grammar, should ensure repeatability of malware description when used to characterize the same malware instance with the same techniques. This should be especially true at the lowest level; after all, there is only one way to describe the insertion of a registry key and similar attributes.

However, at the higher levels, analysts may classify things differently, depending on their expertise, the observed interaction of the malware, and other factors. Therefore, DADM and its core components should be implemented as explicitly as possible, to remove any potential guesswork and subjectivity from malware description.

REFERENCES

[1] Jan Hoffman, Online bullies pull schools into the fray, The New York Times, June 27, 2010.
[2] Conficker – Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Conficker, 2009.

[3] Anti-virus Comparative Malware Removal Test. http://www.avcomparatives.org/images/stories/test/removal/avc_removal_2009.pdf, 2009.

[4] Holz, T. Honeyblog: Threadgraphs for Visualizing Malware Behavior. http://honeyblog.org/archives/34-Thread-Graphs-for-Visualizing-Malware-Behavior.html, 2009.

[5] Crandall, J., Ensafi, R., Forrest, S., et. Al. The Ecology of Malware. NSPW‟08, 2008. http://www.cs.unm.edu/~crandall/nspw08final.pdf.
[6] Symantec Internet Security Threat Report: 2008 Trends http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xiv_04-2009.en-us.pdf, 2009.

[7] Jennifer Steinhauer, Verdict in MySpace suicide case, The New York Times, November 27, 2008.
[8] Miguel Helft, Tanzina Vega, Retargeting ads follows surfers to other sites, The New York Times, August 29, 2010.
[9] Claire Miller, Jenna Wortham, Technology aside, most people still decline to be located, The New York Times, August 29, 2010.
[10] Thomas Claburn, Facebook promises stronger privacy controls, InformationWeek, April 27, 2009a.
[11] Tracking Ghostnet: Investigating a Cyber-Espionage Network. http://www.scribd.com/doc/13731776/Tracking-GhostNet-Investigating-a-Cyber-Espionage-Network, 2009.

[12] Prince, B. Symantec Sees rise in USB-Based Malware as Reports of U.S. Army Ban Surface. http://www.eweek.com/c/a/Security/Symantec-Sees-Rise-in-USBBased-Malware-as-Reports-of-US-Army-Ban-Surface/, 2008.

[13] F-Secure Reports Amount of Malware Grew by 100% During 2007. http://www.f-secure.com/en_EMEA/about-us/pressroom/news/2007/fs_news_20071204_1_eng.html, 2007.

[14] Moscaritolo, A. Most Malware Dies Within 24 Hours. SC Magazine. http://www.scmagazineus.com/most-malware-dies-within-24-hours/article/146384/, 2009.

[15] Kruegel, C., Robertson, W., and Vigna, G. Detecting Kernel-level Rootkits Through Binary Analysis. Proceedings of the 20th Annual Computer Security Applications Conference, 91-100, 2004. DOI=http://dx.doi.org/10.1109/CSAC.2004.19

[16] Stinson, E., and Mitchell, J. Characterizing Bots' Remote Control Behavior. Proceedings of the 4th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment, 89-108, 2007.

[17] Christodorescu, M., Jha., S., and Kruegel, C. Mining Specifications of Malicious Behavior. Proceedings of the 6th joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering, 2007.

[18] Fukushima, Y., Akihiro, S., Yoshiaki, H., and Kouichi, S. Malware Detection Focusing on Behaviors of Process and its Implementation, Joint Workshop on Information Security JWIS2009), 2009.

[19] Vigna, G. Static Disassembly and Code Analysis. Advances in Information Security: Malware Detection, 27, 2007. DOI= http://dx.doi.org/10.1007/978-0-387-44599-1_2

[20] Bayer, U., Moser, A., Kruegel, C., and Kirda, E. Dynamic Analysis of Malicious Code. Journal of Computer Virology, 2:67-77, 2006. DOI= http://dx.doi.org/10.1007/s11416-006-0012-2

[21] Moore, A. P.; Ellison, R. J.; & Linger, R. C. Attack Modeling for Information Security and Survivability (CMU/SEI-2001-TN-001, ADA388771). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001.

[22] Messmer, E. Blended Security Threats on the Rise, IBM Says. http://www.infoworld.com/d/security-central/blended-security-threats-rise-ibm-says-346, 2008.

[23] Bontchev, V. Current Status of the CARO Naming Scheme. http://www.people.frisk-software.com/~bontchev/papers/naming.html, 2009.

[24] Gordon, S. That Which We Call a Rose.A. Virus Bulletin, 2003.

[25] Common Malware Enumeration (CME). http://cme.mitre.org/index.html, 2009.

[26] Vamosi, R. Security Watch: Taking the Internet by Storm. http://reviews.cnet.com/4520-3513_7-6725188-1.html?tag=nl.e404, 2007.

[27] IEEE Malware Working Group Wiki. http://ieee.sanasecurity.com/wiki/index.php/Main_Page, 2009.

[28] Zeitser, L. Categories of Common Malware Traits. http://isc.sans.edu/diary.html?storyid=7186, 2009.

[29] Hines, M. Malware Boom Puts Pressure On Second-Tier AV Labs. http://www.infoworld.com/d/security-central/malware-boom-puts-pressure-second-tier-av-labs-882, 2007.

[30] Anti-Spyware Coalition. http://www.antispywarecoalition.org/index.htm, 2009.

[31] IEEE-SA Industry Connections Malware Working Group. http://standards.ieee.org/prod-serv/indconn/icsgmal/index.html, 2009.

[32] Kerner, S. Researchers: Conficker Still an Active Threat. http://www.internetnews.com/security/article.php/3832846, 2009.

[33] An Analysis of Conficker. http://mtc.sri.com/Conficker/, 2009.

[34] Win32/Conficker.B – CA. http://www.ca.com/securityadvisor/virusinfo/virus.aspx?id=76852, 2009.

[35] Leder, F., and Werner, T. Know Your Enemy: Containing Conficker. http://www.honeynet.org/papers/conficker, 2009.

[36] Fitzgibbon, N. and Wood, M. Conficker.C A Technical Analysis. http://www.sophos.com/sophos/docs/eng/marketing_material/conficker-analysis.pdf, 2009.

[37] SANS ISC Conficker Diaries. http://isc.sans.edu/tag.html?tag=conficker, 2009.

[38] The Downadup Codex, Edition 2.0. http://www.symantec.com/connect/blogs/downadup-codex-edition-20, 2009.

[39] WonKim, Trends in the uses of the Internet—2006, Journal of Object

Technology March, 2006.
[40] Won Kim, The negative byproducts of the Internet (keynote paper),
in: Proceedings of the APWeb 2010, April 6–8, 2010, Busan, Korea.
[41] Andrew Kramer, E-mail spam falls after Russian crackdown,
The New York Times, October 26, 2010.
[42] Thomas Claburn, Google sees less spam, more viruses, Information Week,
October 19, 2010.
[43] Riva Richmond, Building an online bulwark to fend off identity
fraud, The New York Times, November 19, 2009.
[44] /http://en.wikipedia.org/wiki/Internet_fraudS.
[45] /http://www.crimes-of-persuasion.com/Crimes/Telemarketing/
Inbound/MinorIn/cramming.htmS.
[46] Jenna Wortham, Myspace turns over 90,000 names of registered sex offenders, The New York
Times, February 4, 2009.
[47] /http://en.wikipedia.org/wiki/Cyber-bullyingS.
[48] Christopher Maag, When the bullies turned faceless, The New York
Times, December 16, 2007.
[49] Thomas Claburn, Google fights street view ban in Switzerland, Information Week, August 24,
2009b.
[50] Saul Hansell, Google executives face jail time for Italian video,
The New York Times, February 2, 2009.
[51] Eric Pfanner, Europe unleashes online gambling to fill coffers,
The New York Times, July 27, 2010.
[52] Brad Stone, Policing the Web's lurid precincts, The New York Times,
July 19, 2010.
[53] Commtouch Software Ltd., Commtouch Reports Spam Trends for
First Half of 2010, Press release, /http://www.commtouch.com/
download/1679S, 2010.
[54] Commtouch Software Ltd., Q1 2010 Internet Threats Trend Report,
/http://www.Commtouch.com/download/1679S, 2010.
[55] Terry Zink, How Much Do Spammers Actually Make? /http://blogs.
msdn.com/b/tzink/archive/2008/08/28/how-much-do-spammers-actually-make.aspxS.
[56] Digital Trends, How Do Spammers Make Money? /http://www.
digitaltrends.com/computing/how-do-spammers-make-money/S.
[57] Tom Spring, Spam Slayer: Slaying Spam-Spewing Zombie PCs, PC
World, 2005-06-20.
[58] /http://www.crn.com/security/212002220;jsessionid=GQY4WNGECJ
V4NQE1GHPSKHWATMY32JVNS.
[59] /http://en.wikipedia.org/wiki/E-mail_spamS.
[60] /http://www.opus1.comS.
[61] /http://www.opus1.com/www/whitepapers/antispamfeb2007. pdfS.
[62] /http://www.McAfee.comS.

[63] /http://www.kaspersky.comS.
[64] I.J. Androutsopoulos, K. Chandrinos, G. Paliouras, C. Spyropoulos, An evaluation of naı¨ve Bayesian anti-spam filtering, in: Proceedings of the Workshop on Machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML 2000), 2000.
[65] H. Drucker, D. Wu, Support vector machines for spam categorization, IEEE Transactions on Neural Networks 10 (5) (1999).
[66] O. Boykin, V. Roychowdhury, Personal e-mail network: an effective anti-spam tool, Arxiv preprint cond-mat/0402143, 2004—/arxiv.orgS, 2004.
[67] /http://en.wikipedia.org/wiki/HITS_algorithmS.
[68] /http://en.wikipedia.org/wiki/SpamdexingS.
[69] Z. Gyongyi, H. Garcia-Molina, J. Pedersen, Combating Web spam with TrustRank, in: Proceedings of the 30th International Conference on Very Large Data Base, 2004.
[70] A.A. Benczur, K. Csalogany, T. Sarlos, M. Uher, SpamRank-fully automatic link spam detection work in progress, in: Proceedings of the First International Workshop on Adversarial Information Retrieval on the Web, 2005.
[71] B. Wu, B. Davison, Identifying link farm spam pages, in: Proceedings of the International World Wide Web Conference, 2005.
[72] D. Fetterly, M. Manasse, M. Najork, Spam, damn spam, and statistics: using statistical analysis to locate spam web pages, in: Proceedings of the Seventh International Workshop on theWeband Databases, June 2004.
[73] P. Kolari, A. Java, T. Finin, Characterizing the Spologsphere, in: Proceedings of the Third Annual Workshop on World Wide Web, May 2006.
[74] /http://www.hubspot.com/productsS.
[75] John Markoff, Defying experts, rogue computer code still lurks, The New York Times, August 27, 2010.
[76] John Markoff, Malicious software infects computers, The New York Times, February 19, 2010.
[77] Brian Knowlton, Military computer attack confirmed, The New York Times, August 25, 2010.
[78] /http://en.wikipedia.org/wiki/Trojan_horse_(computing)S.
[79] /http://en.wikipedia.org/wiki/Computer_virusS.
[80] /http://en.wikipedia.org/wiki/Computer_wormS.
[81] Security firm: MyDoom worm fastest yet CNN.com, Time Warner, 2004-01-28.
[82] /http://en.wikipedia.ogr/wiki/SpywareS.
[83] Hua Yuan, Guoging Chen, Junjie Wu, Hui Xiong, Towards controlling virus propagation in information system with point-to-group information sharing, Decision Support Systems 48 (1) (2009).
[84] /http://www.microsoft.com/security/malwaremoveS.
[85] Oleg Zaytesv, Rootkits, Spayware/Adware, Keyloggers and Backdoors: Detection and Neutralization, Spyware/Adware, Keyloggers and Backdoors: Detection and Neutralization, A-List Publishing, 2006.
[86] /http://www.ahnlab.comS.
[87] /http://www.symantec.comS.
[88] /http://www.activevirusshield.comS.
[89] /http://www.zonealarm.comS.
[90] /http://www.f-secure.comS.

[91] /http://www.tgsoft.itS.
[92] /http://personalfirewall.comodo.comS.
[93] /http://www.brothersoft.comS.
[94] /http://www.misec.netS.
[95] /http://www.simplysup.comS.
[96] /http://www.moosoft.comS.
[97] /http://www.lavasoft.comS.
[98] /http://www.sofer-networking.orgS.
[99] /http://www.windowsregistrysoftware.comS.
[100] /http://www.Shine-Star.com.cnS.
[101] AhnLab ASEC Report , 2008 /http://www.ahnlab.comS.
[102] John Aycock, Spyware and Adware, first ed., Springer Publishing Company, 2009.
[103] J. Murphy, P. Elmer-Dewitt, M. Krance, Computers: the 414 gang strikes again, Time Magazine, August 1983.
[104] K. Higgins, ‘Aurora’ attacks still under way, investigators closing in on malware creators, DarkReading, February 2010.
[105] D.J. Bernstein, SYN cookies, /http://cr.yp.to/syncookies.htmlS, 1996.
[106] P. Ferguson, D. Senie, Network ingress filtering: defeating denial of service attacks which employ IP source address spoofing, RFC 2827, Internet Engineering Task Force (IETF), 2000.
[107] K. Park, H. Lee, On the effectiveness of router-based packet filtering for distributed DoS attack prevention in power-law Internets, in:Proceedings of the ACM SIGCOMM, 2001, pp. 15–26.
[108] T.M. Gil, M. Poleto, MULTOPS: a data-structure for bandwidth attack detection, in: Proceedings of the 10th Usenix Security Symposium, August 2001.
[109] K. Scarfone, P. Mell, Guide to intrusion detection and prevention systems (idps), Technical Report, NIST, National Institute of Standards and Technology, U.S. Department of Commerce, 2007.
[110] A. Josang, B. AlFayyadh, T. Grandison, M. AlZomai, J. McNamara, Security usability principles for vulnerability analysis and risk assessment, in: Proceedings of the Annual Computer Security Applications Conference, 2007.
[111] /http://www.mozilla.com/en-US/firefox/phishing-protectionS.
[112] /http://macmost.com/safari-32-anti-phishing-protection.htmlS.
[113] /http://www.ie-vista.com/phishing.htmlS.
[114] /http://www.opera.com/docs/fraudprotectionS.
[115] /http://chrome.blogspot.com/2009/04/google-chromes-univer sal-terms.htmlS.
[116] J. Kirk, Phishing scam takes aim at MySpace.com, PCWorld, June 2,
2006.
[117] R. Rasmussen, G. Aaron, Global phishing survey: trends and domain name use in 2H2009, Anti-Phishing Working Group, May 2010.
[118] Phishing attack victims likely targets for identity theft, Gartner, May 4, 2004.
[119] Gartner says ATM/debit card fraud resulted in $2.75 billion in losses in past year, Gartner, August 2, 2005.
[120] Gartner survey shows phishing attacks escalated in 2007; more than $3 billion lost to these attacks, Gartner, December 7, 2007.
[121] Gartner says number of phishing attacks on U.S. Consumers increased 40 percent in 2008, Gartner, April 14, 2009.
[122] C. Herley, D. Florencio, A profitless endeavor: phishing as tragedy of the commons, new security paradigms workshop, September 2008.

[123] /http://en.wikipedia.org/wiki/PhishingS.
[124] E. Gabrilovich, A. Gontmakher, The homograph attack, Communications of the ACM 45 (2) (2002) 128.
[125] E. Johanson, The state of homograph attacks Rev 1.1. The Shmoo Group, February 11, 2005.
[126] J. Leyden, Cybercrooks lurk in shadows of big-nameWeb sites, The Register, December 12, 2007.
[127] /http://www.owasp.org/index.php/Open_redirectS.
[128] Cross site scripting techniques and mitigation, GovcertUK, October 2007.
[129] /http://www.azarask.in/blog/post/ a-new-type-of-phishing-attackS.

APPENDIX A: GLOSSARY
A.I: General Terminology

1) **Artifact:** an entity remaining on a system after the execution of malware that is the result of state changes brought on through its insertion, infection and payload. In terms of dynamic malware analysis, this can be thought of as the difference in system state before and after malware execution. Examples: file system objects (i.e. files, directories), registry keys, network ports, etc.
2) **Attribute:** a characteristic of malware that can be used as a descriptor. For DADM, attributes can include low-level observables, mid-level behaviors, the categories of the high-level taxonomy, and metadata.
3) **Attack Pattern:** a description of a common method for exploiting software, which can include the attacker's perspective and guidance on ways to mitigate their effect. Example: Exploiting incorrectly configured SSL security levels.
4) **Behavior:** the end result of the execution of a specific set of instructions by malware. In this manner, a behavior can be thought of as the consequence of an action. Example: the consequence of inserting a registry key (action) is allowing malware to become resident at system start-up (behavior).
5) **Botnet:** the network of infected computers (or „bots‟) created by worms and other forms of malware with embedded command & control mechanisms. Commonly used to send email spam, as well as originators of DDOS attacks.
6) **Entrenchment:** the subjective degree of how well-established and deeply-rooted a malware instance is designed to become on a system. This is most relevant to remediation, in the sense that a malware instance with a high degree of entrenchment (e.g. a rootkit) is generally more difficult to completely remove than an instance with a lesser degree of entrenchment.
7) **Metadata:** any malware data that by itself is not a distinct attribute. In this manner, metadata can be thought of information capable of more fully describing malware attributes.
8) **Observable**: any data that can be obtained by observing the host-level execution of malware on a system through some form of instrumentation. It is typically obtained through dynamic analysis methods and is dependent on the host operating system. This is the overarching category which artifacts fall under, but it also includes dynamic entities such as the processes spawned and network connections initiated by malware. Examples (broad): file system changes, GUI events, etc.
9) **Remediation:** the process of cleaning up and reverting back to a non-malicious system state after a malware infection. This can involve the removal of malicious binaries, registry keys, and other artifacts, as well as the reversion of artifacts not created by the malware to their state before the infection.
10) **Sandbox:** a behavioral/dynamic malware analysis system that can replicate most of the functionally of a target host while still maintaining a layer of separation between non-sandboxed machines and networks.
11) **Side-effect:** an observable that refers to the unintentional or unforeseen interaction between a malware instance and a particular software environment. For instance, Conficker.B‟s Windows network share propagation mechanism utilizes a brute-force password guessing behavior that has the potential to lockout user accounts based on group policy.
12) **Type:** any of the AV/security community's commonly used monikers for groupings of malware that share some common characteristic. Examples: virus, trojan, worm, backdoor, keylogger, rootkit, bot, etc.

13) **Variant:** a malware instance that shares some code, and therefore behaviors, with another instance or family of malware.
14) **Vector:** the specific method that malware uses to propagate itself. Examples: software vulnerability exploitation, social-engineering, etc.

A.II: High-level Malware Processes and Mechanisms [24]

1) **Armoring** (mechanism)**:** a mechanism employed by malware for the purpose of impeding its analysis. Such mechanisms are typically targeted at specific analysis methods. This is potentially one of the categories in DADM‟s high-level taxonomy. Examples: binary packing, anti-debugging mechanisms, virtual machine checks (dynamic analysis), etc.
2) **Command & Control** (mechanism)**:** a mechanism employed by malware for the purpose of executing commands specified by a remote attacker. Such mechanisms can be used for controlling a single, target machine to multitudes of "zombies" as in botnets. Examples: IRC-based, fast-flux, domain-generation, etc.
3) **Exfiltration** (process)**:** a mechanism employed by malware for the purpose of collecting and then transporting valuable information off of a system. While this is typically accomplished by uploading the data to a remote server, it can also be achieved by saving the data to an encrypted archive for physical retrieval. This is potentially one of the sub-categories of payload.
4) **Infection** (process)**:** the process by which malware instantiates or "installs" itself on a system; typically preceded by insertion. This is potentially one of the categories in DADM‟s high-level taxonomy. Examples: writing a file to some directory, injecting a malicious binary into a process‟ address space, executing a malicious process, etc.
5) **Insertion** (process)**:** the process and vector used by malware for gaining initial entry into a system. This and infection are the only mechanisms common to all malware. This is potentially one of the categories in DADM‟s high-level taxonomy. Examples: software vulnerability exploitation, social-engineering, etc.
6)**Metamorphism:** a mechanism used by malware for automatically re-coding itself each time it propagates or is otherwise distributed, primarily for the purpose of evading detection through physical signatures.
7) **Obfuscation** (mechanism)**:** a mechanism employed by malware for concealing its presence on a system. This is potentially one of the categories in DADM‟s high-level taxonomy. Examples: utilizing the same name as a benign process, changing the last modified date of a malicious binary to that of a known binary, etc.
8) **Persistence** (process)**:** a process by which malware ensures continual execution on a system, independent of low-level system events such as shutdowns and reboots. This is potentially a sub-category of infection. Examples: insertion of an auto-run registry key, installation of a malicious binary as a system service, etc.
9) **Payload:** the specific malware attributes unrelated to insertion, infection, armoring, obfuscation, and self-defense. Therefore, a malware's payload can be thought of as the actions performed after the successful infection of a system, and are directly tied into the purpose behind the malware. This is potentially one of the categories in DADM‟s high-level taxonomy.
10) **Polymorphism:** a mechanism employed by malware for the purpose of changing the appearance of its run-time code; some common methods include encryption and appending data. Like polymorphism, this is primarily intended to evade signature-based detection techniques.
11) **Propagation** (mechanism)**:** a mechanism utilized by malware for the purpose of spreading to other machines. This is potentially a sub-category of payload.

12)**Self-Defense** (mechanism)**:** a mechanism employed by malware that is intended to inhibit its removal after the successful infection of a system. This is potentially one of the categories in DADM"s high-level taxonomy. Examples: the removal/shutdown of AV products, the disabling of specific services, etc.