Assessing the Intentions and Timing of Malware

Brent Maheux

"*Bien mal acquis ne profite jamais.*"

(Ill-gotten gains seldom prosper.)

French proverb

Malware has become a significant, complex, and widespread problem within the computer industry. It represents one of the most prevalent threats to cybersecurity and is increasingly able to circumvent current detection and mitigation techniques. To help better understand when a malware attack might happen, this article proposes an intention-based classification of malware and merges it with an optimal timing model to help predict the timing of malware based on its classification. The classification model is based on an examination of eight malware samples, and it identifies four malware classifications and commonalities based on the dimensions of persistence and stealth. The goal of the article is to provide a better understanding of when cyber-conflict will happen, and to help defenders better mitigate the potential damage.

Introduction

In today's online environment, computer systems now dominate our personal, business, and financial lives. However, our dependency on these systems also makes us vulnerable to cybercriminals. The cost of cybercrime now exceeds $110 billion USD and affects 566 million victims annually, which equates to 1.5 million victims per day or 18 victims per second (Semantec, 2012). Malware, which is short for "malicious software" and includes computer viruses, worms, trojan horses, and spyware (TechTerms, 2014), which are used for a range of illicit activities such as distributing spam email and stealing sensitive information.

Although there has been a lot of research on detecting malware (e.g., Baecher et al., 2006; Gu et al., 2007; Invernizzi et al., 2014; Jain & Bajaj, 2014; Jiang et al., 2007; Peng et al., 2013) and analyzing it from a technical perspective (e.g., Dinaburg et al., 2008; Jain & Bajaj, 2014; Moser et al., 2007; Willems et al., 2007; Yin et al., 2007), there is a lack of research on timing and categorizing malware based on its intentions. A greater understanding of the intentions of attackers will increase the defender's knowledge on how to mitigate attacks.

This article examines an evolutionary timeline of malware based on eight examples of malware dating from the first computer virus in 1971 (Gatto, 2011) through to a recent example from 2012. These examples are used to develop an intention-based classification of malware, which is then combined with Axelrod and Iliev's (2013) optimal timing model. The optimal timing model deals with the question of when the malware should be used given that its use today may well prevent it from being available for use later. The optimal timing model is presented from the perspective of the offense – helping predict the best time to use a resource. However, the results are equally relevant to a defender who wants to estimate how high the stakes have to be in order for the offense to use their resource. When the optimal timing model is combined with the intention-based classification, the new model helps clarify how the timing of malware can depend on the stakes involved in the present situation, as well as the characteristics of the resource for exploitation. Even further, the model helps predict the level of sophistication one could be facing, increasing the chances of mitigating the malware (Galarneau, 2002; Mell et al., 2005; Symantec, 2014).

Axelrod and Iliev test their optimal timing model on four individual case study examples. Combining the model on a broader class of malware samples will further test their model or allow new perspectives and theories to evolve. Because both models use the same definitions for a malware's stealth and persistence capabilities, they can be easily combined to provide a better understanding of the intentions and timing of the attacker's malware.

Assessing the Intentions and Timing of Malware

Brent Maheux

This article is structured as follows. The first section describes and analyzes eight examples of malware, from the first computer virus in 1971 to a case of cyberwarfare in 2012. Next, Axelrod and Iliev's (2013) optimal timing model is introduced and applied to the context of malware. Then, drawing upon the examples of malware analyzed earlier, an intention-based classification of malware is proposed and combined with the optimal timing model to illustrate how the optimal timing of malware can be determined depending on the attacker's intentions. The final section provides conclusions.

Examples of Malware

In this section, eight examples illustrate the evolution of malware, ranging from the first experimental computer virus from 1971 to a cyberespionage application that was discovered in 2012. These eight cases were selected as being noteworthy examples of malware based on a combination of timelines (Hansen, 2013; Infoplease, 2012; Khanse, 2014; Larsen, 2012; Malware Database, 2014; PC History, 2003; Standler, 2008). The eight examples are spread out over the history of malware and are generally representative of contemporary malware examples.

1. *Creeper:* The first virus. In 1971, the Creeper system, now considered to be the first computer virus, was an experimental self-replicating program that infected DEC PDP-10 computers running the TENEX operating system (Gatto, 2011). Creeper gained access via the ARPANET by searching for a machine within the network, transferring itself, displaying a message, then starting over, thereby hopping from system to system. It was developed for experimental purposes, as a proof of concept within an academic research context.

2. *Elk Cloner:* The first outbreak. Elk Cloner was created in 1982 as a prank by a 15-year-old high school student. The virus attached itself to the operating system of Apple II computers and then spread itself via floppy disk to other computers, on which it would display a poem instead of loading a game. Elk Cloner is one of the first known viruses that spread beyond the computer system or laboratory in which it was written (Rouse, 2005).

3. *Happy99:* The happy worm. As the name suggests, this worm was developed 1999 and usually arrived as an email attachment or new post that was named Happy99.exe. Once executed, Happy99 would display fireworks, then copy itself to the windows system folder and then email itself to all contacts listed on the system. Lacking any destructive payload, Happy99 would not cause damage to the actual affected computer; it was simply a prank (Elnitiarta, 2007).

4. *Code Red:* Vulnerable web servers. In 2001, Code Red infected web servers, where it automatically spread by exploiting a known vulnerability in Microsoft IIS servers. In less than one week, nearly 400,000 servers were infected, and the homepage of their hosted websites was replaced with the message "Hacked By Chinese!" Code Red had a distinguishing feature designed to flood the White House website with traffic from the infected servers, which likely makes it the first case of documented political "hacktivism" on a large scale (Lovet, 2011).

5. *Blaster:* A large prank. In 2003, the Blaster worm spread on computers running the Microsoft operating systems Windows XP and Windows 2000, with damage totaling in the hundreds of millions (Dougherty et al, 2003). It was notable for the two hidden text strings, the first of which said "I just want to say LOVE YOU SAN!" and the second of which was a message to Microsoft CEO Bill Gates.

6. *Zeus:* Malware as a service. Over $70 million USD was stolen from users who were infected with the Zeus malware. It was one of the first major botnet malware applications that would go undetected by updated antivirus and go unnoticed by people who were using infected computers. Zeus was capable of being used to carry out malicious and criminal tasks, often being used to steal banking information. Zeus initially started to infect computers in 2007, and by 2009, security company Prevx discovered that Zeus had compromised over 74,000 FTP accounts on websites of such companies as Bank of America, NASA, Monster.com, ABC, Oracle, Cisco, Amazon, and BusinessWeek (Ragan, 2009).

7. *Stuxnet:* The stealthy one. Discovered in 2010, the Stuxnet virus would propagate across a network, scanning for unique Programmable Logic Controllers (PLCs) and certain software. Once it found the correct machine to reside on, it would infect the machine with a rootkit and start modifying the code, giving unexpected commands to the PLC while returning a loop of normal operating system values to the users. Multiple zero-day exploits were used on an estimated 16,000 computers that were infected by the Stuxnet virus, including Iran's nuclear enrichment plant at Natanz (Emerson, 2012).

Assessing the Intentions and Timing of Malware

Brent Maheux

8. *Flame:* Cyberespionage. Flame is a modular computer malware application discovered in 2012 that attacks computers running the Microsoft Windows operating system. The program is being used for targeted cyberespionage in Middle Eastern countries. Flame can spread over systems through the local area network (LAN) or via USB device and has the ability to record audio, screenshots, keyboard activity, and network traffic. According to estimates by Kaspersky in May 2012, Flame had initially infected approximately 1,000 machines with victims including governmental organizations, educational institutions, and private individuals. In total, Kaspersky estimates more than 5,000 computers were infected (Kaspersky Lab, 2013).

As shown in Table 1, the eight examples of malware can be summarized along the following six dimensions:

1. *Year:* date of first discovery.

2. *Intention:* the reason the malware was created. Types of intentions include experimental (including research, entertainment, demonstrations of skill), financial (including theft and fraud), political (including "hacktivists"), and cyberwarfare (including state-sponsored attacks).

3. *Initial access:* how the malware gained access to the system or network. Means of initial access include social engineering (i.e., psychological manipulation), a zero-day vulnerability (i.e., a previously unknown vulnerability in a computer application), and a known vulnerability.

4. *Stealth:* the probability that, if you use a resource now, it will still be available to use later (Axlerod & Iliev, 2013).

5. *Persistence:* the probability that, if you refrain from using a resource now, it will still be available to use in the future (Axlerod & Iliev, 2013).

6. *Extent:* the number of computers affected.

As Table 1 shows, the number of computers affected by the malware increases over time, except in the recent case of Flame, which is malware for targeted espionage, not widespread impact. Early examples of malware were readily detected and did not persist for long, and tended to rely on known vulnerabilities and social engineering for initial access. Later examples, particularly in malware for cyberwarfare, show a trend toward more targeted attacks with increased stealth and persistence.

Modelling Malware Based on Intentions and Timing

The design and features of a particular malware application will depends on the creator's intentions, and its users must also take into account the optimal timing of its desired impact. In the general context of cybersecur-

Table 1. Examples of malware

Name	Year	Intention	Initial access	Stealth	Persistence	Extent
Creeper	1971	Experimental	Known vulnerability	Low	Low	< 1k
Elk Cloner	1982	Experimental	Social engineering	Low	Low	< 1k
Happy99	1999	Experimental	Social engineering	Low	Low	10k
Code Red	2001	Political	Known vulnerability	Medium	High	400k
Blaster	2003	Experimental	Known vulnerability	Low	Low	8M
Zeus	2007	Financial	Social engineering	Medium	Low	3.6M
Stuxnet	2010	Cyberwarfare	Zero-day	High	High	16k
Flame	2013	Cyberwarfare	Zero-day	High	High	1k

Assessing the Intentions and Timing of Malware

Brent Maheux

ity, Axelrod and Iliev (2013) developed an optimal timing model to help understand when a given attacker should exploit its capacity to do harm. Their model considers important assumptions about the stakes at hand and the resource characteristics in terms of stealth and persistence:

1. *Stakes:* their model assumes that the attacker knows the current stakes of how important the target currently is but does not know what the stakes will be at any future point – although they do know the distribution of stakes over time.

2. *Stealth:* the probability that, if you use a resource now, it will still be available to use later.

3. *Persistence:* the probability that, if you refrain from using a resource now, it will still be available to use in the future.

Thus, Axelrod and Iliev's (2013) optimal timing model can be used to predict the optimal time to maximize the value of a particular malware application if an attacker knows the current stakes and the application's capabilities in terms stealth and persistence. An attack-value threshold can be calculated based on the malware's stealth and persistence and the capacity and vigilance of the intended target. For instance, the stealth of malware used against a well-protected target is likely to be less than the stealth of the same malware against a target that is not particularly attentive to security. Likewise, malware will typically have less persistence against a target that keeps its systems up-to-date with security patches than against a target that does not.

Thus, stealth and persistence depend on both the characteristics of the malware itself and the context of its use. Ideally, the attacker would have security knowledge of the systems they are trying to compromise. In the real world, and in Axelrod and Iliev's (2013) optimal timing model, the characteristics of stealth, persistence, and stakes can be weighted differently. However, for simplicity in this preliminary proposal, the model weighs each of the characteristics the same.

Overall, the optimal timing model predicts the three factors that favour attacker patience: low stealth, high persistence, and low stakes. However, when the stakes are high, the model favours high stealth and low persistence. Indeed, based on the analysis of the cases shown in Table 1, the attacker's intentions can be mapped along the two dimensions of stealth and persistence, as shown in Figure 1.

The political malware examples would be found in the top left corner of Figure 1, which is characterized by high persistence and low stealth. For example, "hacktivist" malware often has high persistence and goes undetected until the group wants to raise awareness of a particular situation (Tarzey & Fernandes, 2013). Cyberwarefare malware uses high stealth and high persistence to stay undetected for as long as possible. Financial malware has high stealth, enabling its creators to steal information through social engineering or misleading users; however, it has low persistence because cases of social engineering often have a limited lifespan because they are often based on current events (Conheady, 2012). The final classification is experimental, with low stealth and low persistence, experimental malware does not persist on computers nor does have a potential lifespan because they are often based off of publicly known weaknesses in a system and are created simply to show how an attacker can take advantage of the weakness. Within the set of malware samples studied in this article, all experimental malware displayed messages indicating that it was on the computer and then it would be deleted by users or the vulnerability would be patched.

The classification shown in Figure 1 can be enhanced by introducing variable stakes, as described in Axelrod and Iliev's (2013) model. Table 2 shows three scenarios of low, constant, and high stakes and the optimal timing for the use of malware depending on its intention. When the stakes are low, the optimal timing model determines that the current time is not the optimal time to use the malware for any malware classification, except, potentially financial malware.

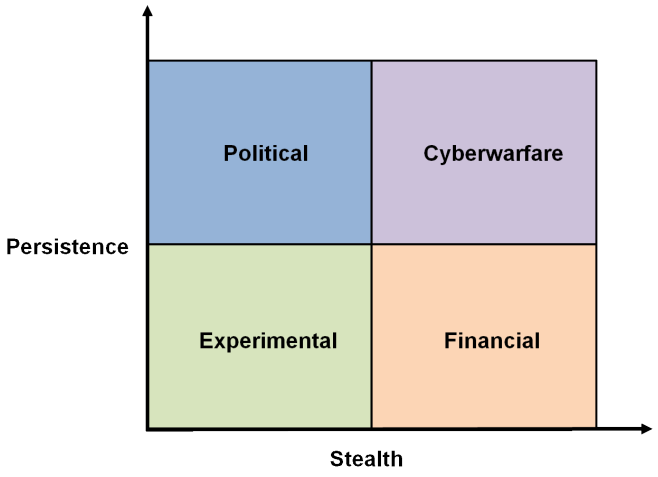

Figure 1. An intention-based classification of malware

Assessing the Intentions and Timing of Malware

Brent Maheux

Under constant stakes, the results in Table 2 show that financial malware should be used immediately. The model suggests the use of financial malware because, as defined by the intention-based classification, financial malware has low persistence and high stealth, making it the exact candidate to use under the optimal timing model. For example, a setting where the stakes are constant over time is the exploitation of stolen credit card information.

Under high stakes, the results in Table 2 show that it is optimal to use the resource immediately, except perhaps when the intention is political. The famous political, or "hacktivist" group, Anonymous, continues to use their resources, but only to send a message relating to a particular event. There is likelihood that they believe their message should be voiced on a particular world event so their stakes are so large that they are willing to sacrifice their resources to do so.

It is important to note the limitations of these results using the same weight for each of the three variables: persistence, stealth, and stakes. In real world examples, and in Axelrod and Iliev's optimal timing model, these values can be weighted differently.

Conclusion

It has been more than 40 years since our first example of malware. Malware evolved, but some of the principles have remained the same. The purposes and motives for malware have changed from educational, protests, and pranks to profit then finally to espionage and sabotage. Intention is an important part of understanding malware; originally, antivirus companies were looking for malware that had financial profit, so many systems were being skipped. Knowing that malware is also being used by governments and military, the search for potential malware activities can be broadened to other poten-

Table 2. The optimal timing of malware use depending on intentions, persistence, stealth, and stakes

Low Stakes	**Intention**	**Persistence**	**Stealth**	**Timing**
	Political	High	Low	Wait
	Cyberwarfare	High	High	Wait
	Experimental	Low	Low	Wait
	Financial	Low	High	---
Constant Stakes	**Intention**	**Persistence**	**Stealth**	**Timing**
	Political	High	Low	Wait
	Cyberwarfare	High	High	---
	Experimental	Low	Low	---
	Financial	Low	High	Use now
High Stakes	**Intention**	**Persistence**	**Stealth**	**Timing**
	Political	High	Low	---
	Cyberwarfare	High	High	Use now
	Experimental	Low	Low	Use now
	Financial	Low	High	Use now

Assessing the Intentions and Timing of Malware

Brent Maheux

tial systems. Understanding the intentions of malware enables the evaluation of the effectiveness of malware defenses.

The concept of initial access has changed slightly over the years. Many of the early examples of malware discussed here needed to be distributed, for instance through email, floppy disk, or USB device, or through a vulnerability in a web service that has an open port. However, the more recent examples – Stuxnet and Flame – were using zero-day exploits. This pattern may be a relatively new trend, because organizations are no longer telling the public or the vulnerable vendors about vulnerabilities; instead they are keeping or selling the techniques (Radianti & Gonzalex, 2007). Again, understanding the purpose of the malware helps in determining how many systems might be affected and how they originally became compromised. If the purpose is financial gain, then it seems likely that many systems will be infected. However, for cyberwarfare, or government-related instances, the examples studied show that only a small, unique set of systems will be infected.

Presented in this article is a model that represents the majority of malware today. The model was created to help understand the potential effectiveness of a malware application's stealth and persistence techniques based on their intentions. And, by combing the optimal timing model by Axelrod and Iliev (2013) with the results of studying the eight malware samples, Table 2 can help predict when an initial attack would likely happen.

About the Author

Brent Maheux is a Senior Software Specialist for the Canadian Government. He holds an MEng degree in Technology Innovation Management from Carleton University in Ottawa, Canada, and a BCS degree in Computer Science from Dalhousie University in Halifax, Canada. He has over 7 years working experience within the public and private sector specializing in product design and implementation.

References

Axelrod, R., & Iliev, R. 2013. Timing of Cyber Conflict. *Proceedings of the National Academy of Sciences of the United States of America,* 111(4): 1298-1303. http://dx.doi.org/10.1073/pnas.1322638111

Baecher, P., Koetter, M., Holz, T., Dornseif, M., & Freiling, F. 2006. The Nepenthes Platform: An Efficient Approach to Collect Malware. *Recent Advances in Intrusion Detection,* 4219: 165-184. http://dx.doi.org/10.1007/11856214_9

Conheady, S. 2012. The Future of Social Engineering. *Privacy PC.* July 17, 2012. http://privacy-pc.com/articles/the-future-of-social-engineering.html

Dinaburg, A., Royal, P., Sharif, M., & Lee, W. 2008. Ether: Malware Analysis Via Hardware Virtualization Extensions. *Proceedings of the 15th ACM Conference on Computer and Communications Security:* 51-62. http://dx.doi.org/10.1145/1455770.1455779

Dougherty, C., Havrilla, J., Hernan, S., & Lindner, M. 2003. W32/Blaster Worm. Historical Advisory CA-2003-20, CERT Division of the Software Engineering Institute. October 1, 2014: http://www.cert.org/historical/advisories/CA-2003-20.cfm

Elnitiarta, R. 2007. Security Response: Happy99.Worm. *Symantec.* October 1, 2014: http://www.symantec.com/security_response/writeup.jsp?docid=2000-121812-3151-99

Emerson, R. 2012. Stuxnet Virus Infected 16,000 Computers, Iran Says. *Huffington Post*, February 18, 2012: http://www.huffingtonpost.com/2012/02/18/stuxnet-virus-iran_n_1286281.html

Galarneau, L. 2002. Anti-virus Software: The Challenge of Being Prepared for Tomorrow's MalWare Today. SANS Institute 2002.

Gatto, K. 2011. The Virus Turns 40. *Phys Org.* November 1, 2014: http://phys.org/news/2011-03-virus.html

Gruener, W. 2012. Kaspersky: Flame Has Three Unidentified Malware Siblings. *Tom's Hardware.* November 1, 2014: http://www.tomshardware.com/news/virus-flame-stuxnet,17644.html

Gu, G., Porras, P., Yegneswaran, V., Fong, M., & Lee, W. 2007. BotHunter: Detecting Malware Infection Through IDS-Driven Dialog Correlation. *Proceedings of the 16th USENIX Security Symposium:* 167-182.

Hansen, P. 2013. History of Malware. *Technology Bell.* November 1, 2014: http://www.technologybell.com/history-of-malware/

Infoplease. 2012. Computer Virus Timeline. *Information Please.* November 1, 2014: http://www.infoplease.com/ipa/A0872842.html

Invernizzi, L., Miskovic, S., Torres, R., Saha, S., Lee, S., Mellia, M. Kruegel, C., & Vigna, G. 2014. Nazca: Detecting Malware Distribution in Large-Scale Networks. Network and Distributed System Security (NDSS) Symposium 2014. February 23, 2014.

Assessing the Intentions and Timing of Malware

Brent Maheux

Jain, M., & Bajaj, P. 2014. Techniques in Detection and Analyzing Malware Executables: A Review. *International Journal of Computer Science and Mobile Computing,* May, 2014 (5): 930–935.

Jiang, X., Wang, X., & Xu, D. 2007. Stealthy Malware Detection through VMM-Based "Out-of-the-Box" Semantic View Reconstruction. *Proceedings of the 14th ACM Conference on Computer and Communications Security:* 128-138. http://dx.doi.org/10.1145/1315245.1315262

Khanse, A. 2014. Evolution of Malware – How It All Began! *The Windows Club*. November 1, 2014: http://www.thewindowsclub.com/evolution-of-malware-virus

Kaspersky Lab, 2013. Who's Spying on You? *Kaspersky Lab*. November 1, 2014: http://media.kaspersky.com/en/business-security/kaspersky-cyber-espionage-whitepaper.pdf

Larsen, C. 2012. A Malware Hall of Fame. *Blue Coat*. November 1, 2014: http://www.bluecoat.com/security/security-archive/2012-10-31/malware-hall-fame

Lovet, G. 2011. 40th Anniversary of the Computer Virus. *Help Net Security*. October 1, 2014: http://www.net-security.org/malware_news.php?id=1668

Malware Database. 2014. Timeline of Noteworthy Computer Viruses, Worms and Trojan Horses. *The Malware Database*. November 1, 2014. http://malware.wikia.com/wiki/Timeline_of_noteworthy_computer_viruses,_worms_and_Trojan_horses.

McDowell, M. 2013. Security Tip (ST04-014): Avoiding Social Engineering and Phishing Attacks. *United States Computer Emergency Readiness Team*. November 1, 2014: https://www.us-cert.gov/ncas/tips/ST04-014

Mell, P., Kent, K., & Nusbaum, J. 2005. *Special Publication 800-83: Guide to Malware Incident Prevention and Handling.* Gaithersburg, MD: Nation Institute of Standards and Technology.

Moser, A., Kruegel, C., & Kirda, E. 2007. Exploring Multiple Execution Paths for Malware Analysis. *Proceedings of 2007 IEEE Symposium on Security and Privacy:* 231-245. http://dx.doi.org/10.1109/SP.2007.17

PC History. 2003. The History of the PC Virus. *PC History*. November 1, 2014: http://www.pc-history.org/pc-virus.htm

Peng, W., Li, F., Zou, X., & Wu, J. 2013. Behavioral Malware Detection in Delay Tolerant Networks. *IEEE Transactions on Parallel and Distributed Systems,* 25(1): 53–63. http://dx.doi.org/10.1109/TPDS.2013.27

Ragan, S. 2009. ZBot Data Dump Discovered with over 74,000 FTP Credentials. *The Tech Herald*. November 1, 2014: http://www.thetechherald.com/articles/ZBot-data-dump-discovered-with-over-74-000-FTP-credentials/6514/

Rouse, M. 2005. Elk Cloner. *SearchSecurity.com*. October 1, 2014: http://searchsecurity.techtarget.com/definition/Elk-Cloner

Semantec. 2012. 2012 Norton Cybercrime Report. Mountain View, CA: Symantec Corporation.

Semantec. 2014. *Preparing for Future Attacks*. Mountain View, CA: Symantec Corporation.

Standler, R. 2008. Examples of Malicious Computer Programs. *Website of Dr. Ronald B. Standler*. November 1, 2014: http://www.rbs2.com/cvirus.htm

Tarzey, B., & Fernandes, L. 2013. The Trouble Heading for Your Business. *Quocirca,* February 2013

TechTerms. 2014. Malware. *TechTerms.com*. November 1, 2014: http://www.techterms.com/definition/malware

Willems, C., Holz, T., & Freiling, F. 2007. Toward Automated Dynamic Malware Analysis Using CWSandbox. *IEEE Security & Privacy,* 5(2): 32-39. http://dx.doi.org/10.1109/MSP.2007.45

Yin, H., Song, D., Egele, M., Kruegel, C., & Kirda, E. 2007. Panorama: Capturing System-Wide Information Flow for Malware Detection and Analysis. *Proceedings of the 14th ACM Conference on Computer and Communications Security:* 116-127. http://dx.doi.org/10.1145/1315245.1315261

Citation: Maheux. B. 2014. Assessing the Intentions and Timing of Malware. *Technology Innovation Management Review,* 4(11): 34–40. http://timreview.ca/article/848

Keywords: malware, cybersecurity, optimal timing, stealth, persistence