Editors: Dino A. Dai Zovi, ddz@theta44.org
Alexander Sotirov, alex@sotirov.net

A Case Study of Intelligence-Driven Defense

Dan Guido
iSEC Partners

MITRE's Common Vulnerability and Exposures (CVE; http://cve.mitre.org) project has identified and tracked more than 8,000 vulnerabilities over the past year. The prevailing wisdom in the security industry and academia is that these vulnerabilities create massive exposure for organizations and must be eradicated. Organizations devote entire teams to ensuring that these vulnerabilities are appropriately handled and typically purchase an array of expensive products to help them do so. Despite these efforts, the number of reported data breaches has dramatically risen in 2011, and the number of identified vulnerabilities continues to grow.

Instead of accepting the status quo, consider the massive proliferation of vulnerabilities from an attacker's perspective. In 2010, how many vulnerabilities were exploited to install SpyEye, Zeus, Gozi, Clampi, and other information-stealing Trojans in massive exploitation campaigns? Only 13. In 2009, that number was 14. This trend holds as far back as 2005, when these types of attacks first emerged. Analysis of attacker data, and a focus on vulnerabilities exploited rather than vulnerabilities discovered, might yield more effective defenses.

This shift is difficult for information security professionals to make. As an industry, we tend to look at architectures, implementations, and the aggregated potential exploitation of an unmanageable number of vulnerabilities. On the other hand, we ignore that there are distinct groups of attackers, each having unique goals and having invested in a business process that might be difficult to change (see Figure 1). Here, I focus on the group that affects the most people: mass malware. In my years working for and consulting with large organizations, I have yet to come across any that are unaffected by it.

Defenses against mass malware have focused on vulnerability mitigation, through patching, intrusion detection systems, and writing and purchasing more secure code, and on malware identification, mainly through antivirus software. Instead, let's approach mitigating this threat by understanding the techniques, tactics, and procedures that make it unique.

The Attacker's Perspective

From the perspective of someone running a mass-malware campaign, seven steps are necessary to steal banking and other credentials from millions of computers around the world:

1. *Gain exposure.* Gain widespread exposure to a large group of Web browsers by purchasing advertisements, performing search engine optimization, and leveraging already compromised users on social networks, and through massive SQL injection campaigns that sometimes compromise more than 1 million websites at a time.[1]
2. *Acquire capabilities.* Acquire the capability to exploit a Web browser by purchasing a *crimeware* pack (also called a Web exploit kit), which contains five to 20 exploits and costs from US$200 to $2,000.
3. *Establish a delivery network.* Host the crimeware pack and establish a series of redirects to obfuscate its location. This is commonly aided by fast- or double-flux networks composed of hundreds of thousands of IP addresses.
4. *Exploit the target.* Deliver the exploit to a website visitor and begin executing malicious code. Usually, this involves exploiting a flaw in a major browser or client-side application such as Internet Explorer or Java.
5. *Install malware.* Download a second stage that installs malware on the target. This malware is typically armored with anti-analysis features, such as repeated encoding and obfuscation by an automatic service to avoid detection by antivirus software.
6. *Establish command and control.* Establish a communications channel with a control server. Interaction with mass malware

1540-7993/11/$26.00 © 2011 IEEE

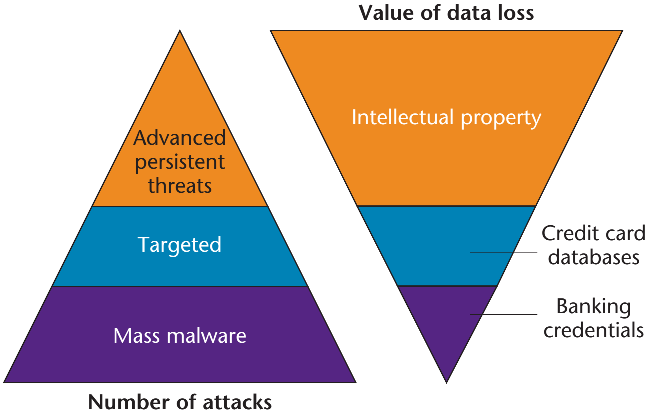

Figure 1. The Internet contains specific groups of threats, each with a different goal.

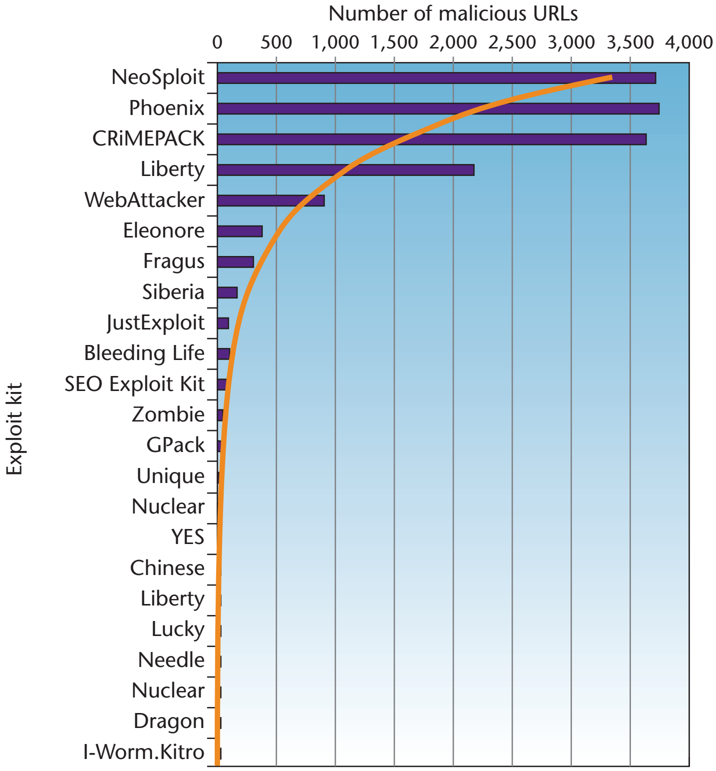

Figure 2. Exploit kit popularity for the first quarter of 2011. The kits' popularity appears to follow an exponential curve. The top four kits were responsible for 85 percent of the volume of malicious URLs, and the top 10 kits were responsible for almost 99 percent of the volume.

is done through scripts and other automated actions owing to the massive number of compromised hosts involved in a single campaign.

7. *Perform actions on objectives.* Collect and exfiltrate banking and other credentials from the compromised victim. Mass malware typically avoids compromising additional systems and moving laterally in the environment.

By laying out these steps, I've developed an *intrusion kill chain* for mass malware. Eric Hutchins and his colleagues first developed and used this concept;[2] I adapted my chain directly from their research.

Enterprises should evaluate kill chains with an eye for capability gaps, high switching costs,[3] and general laziness on the attacker's part. The most naive analysis you can perform with a kill chain is to map the magnitude of effort the attacker has applied at each stage. If you do this for the mass-malware kill chain, the exploitation step clearly stands out. The number of exploited applications is in the tens versus the thousands of potential vulnerabilities, millions of malware samples, and thousands of IPs used in such attacks. This might indicate a capability gap and creates an opportunity for defense. Rather than comprehensively evaluate potential defenses against each step in the kill chain, let's analyze only the exploitation step because it might be a vulnerable point in their intrusion workflow.

Analyzing Exploit Kits

This threat doesn't develop exploits ad hoc to use in mass attacks; rather, it relies on what others have implemented in crimeware packs. First, we need to gather information related to the popularity of those crimeware packs deployed in the wild. Earlier this year, I partnered with ThreatGRID (http://threatgrid.com), a threat intelligence firm, to develop a list mapping each unique malicious URL they observed over the first quarter of 2011 to the name of the exploit kit it hosted.[4] This dataset identifies which kits and how many of them are in use, thus representing a rough picture of exploit kit popularity.

The results were striking (see Figure 2). Exploit kits' popularity appears to follow an exponential curve. The top four kits were responsible for 85 percent of the volume of malicious URLs, and the top 10 kits were responsible for almost 99 percent of the volume. Mass malware is unique in that, owing to its scale of operations, it operates largely in the open, which allows even resource-constrained defenders to acquire copies of most crimeware packs.

To continue my analysis, I collected the source code for the top 15 kits and evaluated each kit's exploitation capabilities. The collected dataset is therefore a mapping of popular crimeware packs to the vulnerabilities they exploit, and it represents the collective capability of mass malware to accomplish this step of the kill chain.

The first conclusion from looking at this data is obvious in retrospect: exploitation focuses on dominant platforms.[5] In the past year, exploits for only five applications developed by four vendors were added to the entire dataset of exploit kits: Microsoft Internet Explorer, Adobe Reader and Flash, Oracle Java, and Apple QuickTime. Each of these applications has significant market share (Internet Explorer has the least), which provided some guarantee that potential victims would be successfully exploited.

The next conclusion from this data is that crimepack authors appear to rely on relatively few information sources. Most of the vulnerabilities abused by the kits over the past two years came from exploit code disclosed in targeted attacks, 0-day disclosures by security researchers with implemented exploit code, and clearly described vulnerabilities disclosed as part of TippingPoint's Zero Day Initiative. If a vulnerability was disclosed in one of these three ways, it was far more likely to be integrated into a crimepack and exploited in mass attacks.

Table 1. Defenses that would have worked on exploits in 2009 and 2010.

Exploit and related defenses	No. of exploits
Memory corruption	19
Defeated by data execution prevention	14
Defeated by address space layout randomization	17
Defeated by the Enhanced Mitigation Experience Toolkit	19
Logic flaws	8
Defeated by not using Java in the Internet zone	4
Defeated by not including EXEs in PDFs	1
Defeated by not using Firefox or Foxit Reader	2

Table 2. The myth of sophistication.

Exploit and developers	No. of exploits
Data-execution-prevention bypass	5
Developed by an advanced persistent threat (APT)	3
Developed by white hats	2
Developed by malware authors	0
Logic flaws	8
Developed by an APT	0
Developed by white hats	8
Developed by malware authors	0

Defense Might Be Easier Than You Think

This dataset allows for more productive analysis. Over the past two years, the kits implemented only 27 exploits. For an organization on 1 January 2009, I identified the decisions it could have made to mitigate all those exploits (see Table 1). Enterprises in 2009 largely ran Windows XP SP3 with Internet Explorer 7, Adobe Reader 9, Office 2007, and a variety of other client-side applications.

Despite the seemingly vast attack surface of modern enterprise desktop systems, protecting them from mass malware requires surprisingly few defensive actions. The overwhelming majority of exploits in the dataset couldn't avoid data execution prevention, and even fewer could bypass address space layout randomization.[6] This dataset had only eight logic flaws, two of which were in software not common on enterprise desktops. Simple configuration changes in Oracle Java and Adobe Reader could have mitigated the remaining six. Defending against these exploits didn't require purchasing any products. Instead, these simple configuration changes are supported by the application vendors and can be centrally managed.

The exploits that bypassed basic memory protection were a point of interest. Had the crimepack authors invested effort into developing these exploits to succeed where others failed? To answer this, I matched each vulnerability to the individual who first published exploit code for it (see Table 2). Neither the exploits that corrupted memory nor the logic flaws originated from a crimepack author. Instead, a dichotomy emerged in which a single group appeared responsible for each of the two classes of flaws. The exploits that corrupted memory were developed predominantly by advanced persistent threats (APTs) and used in targeted attacks. After this code became public, crimepack authors copied it verbatim and used it for mass attacks. The exploits that abused logic flaws originated entirely from security researcher disclosures. In fact, a single researcher had disclosed three of the eight flaws.

This information indicates that two groups are at work: exploit producers and exploit consumers. Crimepack authors survive entirely on others' disclosures and prefer data sources with greater information detail over those with less. An APT's previous operational use of exploit code is the highest indication that this code would also be successful in mass attacks. Similarly, if a prominent security researcher releases exploit code for a severe flaw and it fits into the attacker's operational model, it's far more likely to be used in such attacks. On the other hand, advisories that are ill-described, lack functional exploit code, and don't easily adapt to the attacker's workflow are rarely abused.

This analysis of the attacker data indicates that basic, generic defenses, such as minor reductions of attack surface and using available platform memory protection, are effective against mass malware. This analytical approach has helped identify an optimal, low-cost defensive strategy by identifying and putting pressure on the attackers' gaps in capability. It has also helped avoid defenses that would have little impact on their operational model—namely, trying to identify and fix every vulnerability and writing signatures for malware. This attacker data—the characterization of a group and the knowledge of their operational model—is essential to determine the value of potential defenses. In the end, organizations need to identify what threats they face, what those threats' capabilities are, and how to intelligently defend against those threats. □

References

1. "Update on LizaMoon Mass-Injection and Q&A," Websense, 2011; http://community.websense.com/blogs/securitylabs/archive/2011/03/31/update-on-lizamoon-mass-injection.aspx.
2. E.M. Hutchins et al., "Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," *Proc. 6th Int'l Conf. Information Warfare and Security* (ICIW 11), Academic Conferences Ltd., 2010, pp. 113–125; http://papers.rohanamin.com/wp-content/uploads/papers.rohanamin.com/2011/08/iciw2011.pdf.
3. K. Levchenko et al., "Click Trajectories: End-to-End Analysis of the Spam Value Chain," *Proc. 2011 IEEE Symp. Security and Privacy*, IEEE CS Press, 2011, pp. 431–446; http://cseweb.ucsd.edu/~savage/papers/Oakland11.pdf.
4. D. Guido, "The Exploit Intelligence Project" (PowerPoint presentation), iSEC Partners, 2011; www.isecpartners.com/storage/docs/presentations/EIP-2.0.pdf.
5. StatOwl homepage; http://statowl.com.
6. "Mitigating Software Vulnerabilities," Microsoft, 2011; www.microsoft.com/download/en/details.aspx?displaylang=en&id=26788.

***Dan Guido** is a senior security consultant at iSEC Partners. Contact him at dguido@gmail.com.*