The Hacker Always Gets Through

TJ O'Connor

The Hacker Always Gets Through

T.J. O'Connor

Understanding Exploitation

> *The advent of air power, which can go straight to the vital centers and either neutralize or destroy them, has put a completely new complexion on the old system of making war. It is now realized that the hostile main army in the field is a false objective, and the real objectives are the vital centers.*
>
> Brigadier General William "Billy" Mitchell,
> *Skyways: A Book on Modern Aeronautics* (1930).

In early 2010, security analysts started noticing something really interesting. Attackers from several Chinese hacker groups shifted from a strategy of remote and overt attacks to a stealthy strategy of attacking from compromised pivot points within our own networks. These changing tactics demand significant examination and discussion. Fortunately for us, this strategy allows us to make clearer predictions about the depth of our enemy's exploit arsenal, the ability to develop newer attack vectors, and the adversary's perception of our defensive strategy.

We often refer to this new covert strategy as *watering-hole* attacks. Rather than hunt the entire forest in search of an animal, the hunter waits at the watering hole, knowing the animal will eventually return to drink. For hackers, "watering holes" provide an omnipresent means of aggregating attacks against a specific target set of victims. Consider an attack in May 2013, where members of the Chinese Dark Panda hacking group compromised a Department of Labor (DOL) website, which contained information for members of the Department of Energy (DOE) on handling nuclear materials in the workplace (Blasco, 2013). One could easily assume this attack was not targeted at Department of Labor employees but at Department of Energy nuclear scientists (predictably working in the field of weapons research) who were connected to the DOL website.

Consider the economies of scale that exist in the watering-hole attack strategy. Constant advances in technology can quickly reduce both the cost and the effectiveness of an exploit. In the case of watering-hole attacks, the adversary must essentially *burn a capability* to reach and compromise the intended target. In the specific case of the DOL attack, the attackers implanted an Internet Explorer 8 0-day on the DOL webpage with the full expectation of eventual discovery. However, the hackers predicted this eventual discovery would happen only after the successful attack pivot through the DOL webpage to victims in the Department of Energy. This eventual discovery cost the Chinese hackers roughly $60,000–$75,000 of intellectual property[1] (Miller, 2007). Could a $10,000 investment in the exploit have hidden it from discovery longer? Certainly. The exploit developers could have added better encoding, utilized a novel method for allocating the payload into memory, or added exploit mitigation bypass strategies. Yet they didn't. Why? It is not because they couldn't afford to. For their own survivability in the face of ongoing or future attacks, it is essential for an attacker to always clear the bar with the *least effective technology that will accomplish the mission.* In the case of this attack, one could estimate that the value of the information gained from the nuclear scientists must have exceeded $75,000. If the information value cost anything less, a different, less costly exploit would have to be employed. So what goes into the cost of an exploit?

The Cost Variables of an Exploit

Of what use is decisive victory in battle if we bleed to death as a result of it?

[1] $75,000 for a modern Internet Explorer remote code execution 0-day is grossly underestimated after IRC discussions on underground exploit development markets. However, Miller's (2007) research provides the most current academic reference for this value.

B.H. Liddell Hart, *Paris, or the Future of War*

Several closely woven pieces must be integrated for a successful exploit. First, an attacker must discover a vulnerability and write proof-of-concept code that exploits that vulnerability. This step often proves the most critical and costly piece of any attack. Underground exploit markets sell proof-of-concept exploits for a range of prices, from $10,000 to $250,000 (Miller, 2007). The range in value depends upon variables like exclusivity of the sale, the popularity of the software exploited, the operating system that the exploit will work on, and the relative security posture of the affected software. Popular software from vendors such as Adobe, Microsoft, Google and Apple can fetch high values. However, popularity doesn't necessarily demand high values. Historically vulnerable software such as Oracle's popular implementation of Java may cost less because of the flooded market exploits against Java. Examine Figure 1 for the prizes for the Pwn2Own exploit development contest for 2013 (Pwn2Own, 2013). Software with a particularly insecure history like Java may fetch lesser values, as represented by the 2013 Pwn2Own competition prizes.[2] In contrast, an exploit for the Google Chrome browser may fetch high values based on the difficulty of overcoming several of the security features in the software.

While prestige and honor come from winning Pwn2Own, the prize often fails to meet the market value of the exploit. Furthermore, the contest rules demand the vulnerability be released to the affected company. Thus, exploit developers fail to profit from their exploits once vendors patch the vulnerabilities. In 2012, security researchers with VUPEN discovered a vulnerability for Internet Explorer 10 on the newly released Windows 8 Operating System. Rather than

[2] The prize values represented in Pwn2Own only offer insight about the actual market value. Typically, the market value grossly exceeds Pwn2Own prizes. However, Brian Krebs documented the sale of a Java 0-day to multiple clients for $5,000 each. (Krebs, 2013)

compete at Pwn2Own, VUPEN sold the exploit to several within its nation-state customer base (Halfacree, 2012). An exclusive sale of a Windows 8 Internet Explorer 10 0-day may fetch over half a million dollars, with further revenue collected as the exploit remains undetected.

Vulnerable Software	Operating System	Prize
Google Chrome	Windows 7	$100,000
Internet Explorer 10	Windows 8	$100,000
Internet Explorer 9	Windows 7	$75,000
Mozilla Firefox	Windows 7	$60,000
Safari	Mac OS X Mountain Lion	$65,000
Adobe Reader XI	Windows 7	$70,000
Adobe Flash	Windows 7	$70,000
Oracle Java	Windows 7	$20,000

Figure 1: Prize Values For 2013 Pwn2Own Exploit Development Competition

While the vulnerability discovery and proof-of-concept consumes a large portion of the cost, several other factors must be considered. Newer operating systems contain several exploit mitigation strategies. While an exploit may succeed on older platform such as Windows XP Service Pack 3, it may fail on Windows 8 because of the newer exploit mitigation strategies (Johnson, 2012). A once simple stack-based buffer overflow may have to turn into a complex memory corruption exploit in order to gain the same level of code execution. For the cost-effectiveness of exploit developers, these exploit mitigation bypass techniques often become the company's "secret sauce"—releasing these techniques may compromise the future revenue of previously sold exploits. Consider Figure 2, which illustrates several exploit techniques used against the Microsoft Windows Operating System (Meer, 2010). While it took several years to get initial mitigation strategies in place, recent mitigation strategies have only taken months.

Exploit Technique	Year Developed	Mitigation Strategy	Year Developed
Stack-based buffer overflow	1972	Stack Cookies (/GS Flag)	2003
		NX feature in Data Execution Protection	2004
		Stack Cookies (/GS v2 Flag)	2005
Overwrite EIP, Jump to register	1996[3]	Windows Vista Partial ASLR	2007
		Windows Vista SP1 Full ASLR	2008
SEH Overwrite	2003	/SAFESEH	2003
		Pointer Guard in Data Execution Protection (DEP)	2004
		SEHOP in EMET v1.0	2009
Heap-Spray	2004	Windows XP SP2 Heap Protection	2004
ASLR Partial Overwrite	2007	Windows Vista SP1 Full ASLR	2008
		Mandatory ASLR in EMET v2.0	2010

Figure 2: Timeline of Exploit Mitigation Strategies

Let's discuss a particular technique developed by the Corelan Team. In February 2013, Peter Van Eeckhoute developed a method for placing malicious code into the heap that bypassed all of Microsoft Windows' current mitigation strategies (Eeckhoute, 2013). Rather than selling the method, he posted it to his blog and incorporated the technique in the open-source Metasploit Framework (Sinn3r, 2013). As of June 2013, over 6,000 exploit developers have studied his technique. While Van Eeckhoute's method works today, it's highly unlikely to survive another six months in the wild. EMET 4, due to be released in July 2013, promises methods for defeating

[3] Although earlier reports exist, "Smashing the Stack for Fun and Profit" in Phrack 49 provides the most well-documented use of overwriting EIP and Jumping to an address stored in a register.

Eeckoute's technique. Developing and sustaining exploit mitigation bypass techniques certainly adds to the cost of an exploit.

In some cases, an exploit may use a second stage dropper to keep itself relatively small, agile, and effective. Instead of the exploit carrying the full malicious payload, it contains only a small bit of code that fetches and executes the second stage dropper from a third-party website. Sometimes attackers will use multiple different droppers on various sites, as in the case of Stuxnet. In contrast, an exploit used only a single dropper in recent very successful attack on the Korean banking infrastructure during the Dark Seoul attack of 2013. Setting up and maintaining multiple drop sites and different dropper executables can prove challenging when targeting tens or hundreds of thousands of victims. If only a single drop site is used, a signature can be rapidly developed and distributed to Internet Service Providers (ISPs) to block future downloads. Techniques such as fast-flux or domain-flux can be used to take advantage of the domain naming

Malware	Dropper Strategy	Infected Victims	Duration
Conficker	Five droppers utilizing domain-flux to point to a pool of 50,000 domains across 110 TLDs. (Doyle, 2001)	9–15 million victims	Effectively January 2008 through April 2009
Dark Seoul	1 dropper (Trojan.Jokra)	Shutdown of two South Korean banks and media companies	One day – March 20, 2013
Code Red	Used infected systems to propagate initial payload by scanning the Internet for future victims	250,000 computers (initially 9 hours)	Less than nine hours on July 19, 2001.

Figure 3: Attack Duration As a Product of Drop Strategy

system to tell exploits to fetch droppers from different physical servers. Thus, the physical infrastructure and supporting drop sites can far exceed the initial cost for the exploit. Examining Figure 3, we see several different successful strategies for attack.

Finally, several creative methods may be developed for the exploit or payload to bypass anti-virus or intrusion detection systems. Stuxnet, for example, contained two legitimately signed digital certificates. These valid developer certificates instructed the operating system and anti-virus software that the software came from a legitimate vendor and therefore should be trusted. Certainly stealing a digital certificate signing authority from a reputable vendor could drive up the cost of the exploit. The Flame malware used a previously unknown method for digitally signing executables utilizing a novel method for colliding MD5 hashes. Creating such a collision technique would require several PhDs in advanced mathematics. Thus, it is not surprising that Flame survived several years in the wild before detection. Furthermore, the nation-state behind Flame must have placed a high value on the intellectual property or intelligence gained by using it. In contrast, a simple method of obfuscation, encoding, or packing may be used to bypass anti-virus for a shorter period of time. The Chinese Elderwood Gang hacking group often uses a technique known as 0x95 encoding to obfuscate the payload of their exploits (O'Gorman, 2012). Such a technique may inexpensively and temporarily bypass protection means. However, reusing an encoding technique may prove more costly by revealing the source of the attack. An attacker must consider the cost of attribution in the overall attack.

The Hidden Cost of Attribution

In many cases, an attacker wishes to remain anonymous for the duration and aftermath of the attack. Furthermore, an attacker may even wish to misattribute his or her actions to another group, nation-state, or crime syndicate. Consider the theft of the intellectual property of a US

corporation. If the US government could identify and positively determine the actors stealing intellectual property, they could impose trade sanctions or penalties upon the offending nation. Yet this is a realm where such positive attribution proves difficult. Unskilled malware analysts often falsely classify an attack based on the originating source Internet Protocol (IP) address of the attack, the dropper, or command and control server.

The IP address must be considered as only one variable of the overall attack. Let's examine the recent research of Symantec to gain an appreciation for how many variables must be considered. In 2012, Symantec released a report attributing a campaign of attacks from 2009 to 2012 to a single actor (O'Gorman, 2012). The attacks included eight unique 0-day exploits and targets in five separate countries, and each attack utilized multiple different droppers and command and control servers (O'Gorman, 2012). During the campaign, the attacker's targets ranged from US-based Google to non-governmental organizations in Taiwan to defense contractors in Australia. Attributing such a broad range to a single actor would appear an overwhelming task.

However, consider the link-analysis depicted in Figure 4. The attackers used the CVE-2010-2884 and CVE-2012-1889 0-day exploits to specifically target visitors to Amnesty International's Hong Kong site. CVE-2012-1889 could then be tied to CVE-2012-1875 because the droppers were hosted on the same server and connected to the same command and control servers. Next, Symantec tied CVE-2012-1875 and CVE-2012-0779 together because they used the same encoding schemes, packers, and payloads. Finally, CVE-2012-0779 and CVE-2012-1535 shared the same payloads.

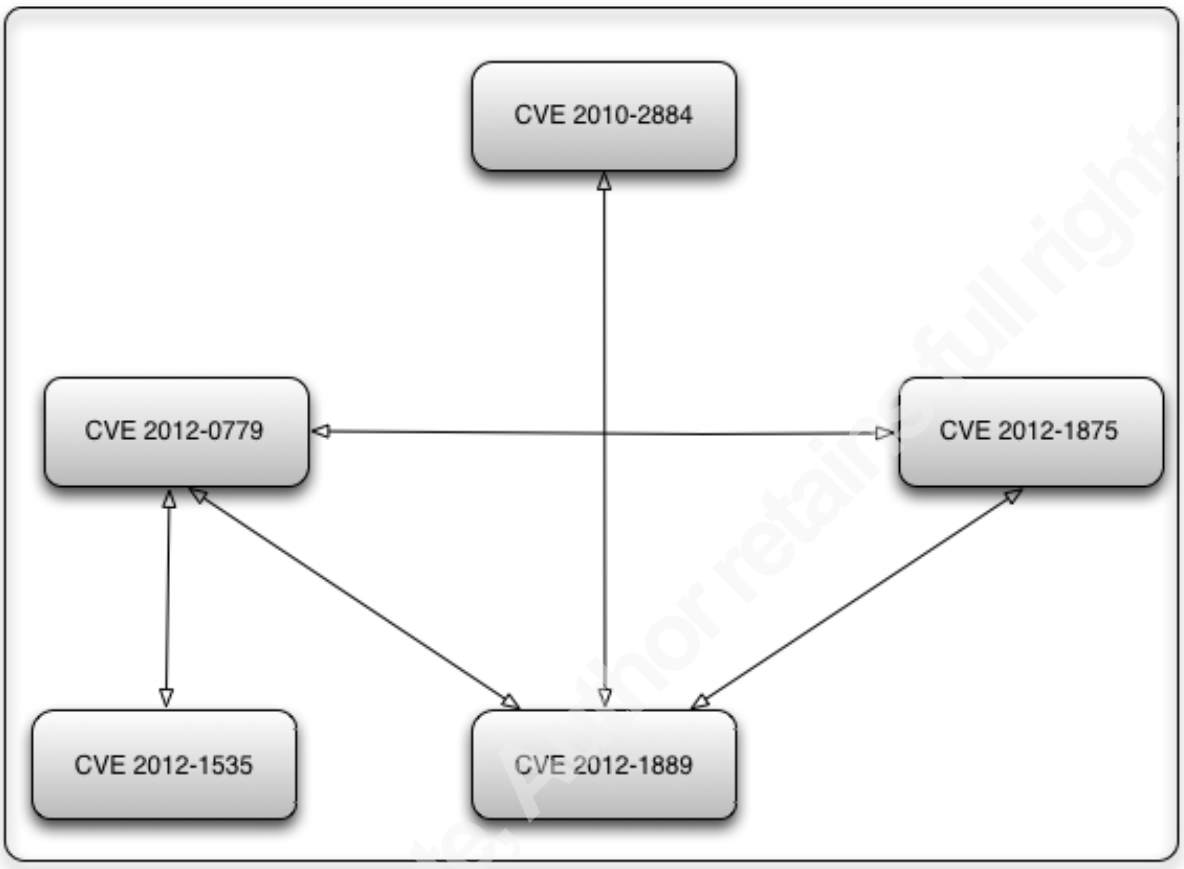

Figure 4: Link Analysis For Elderwood Gang Exploits

Examining the link-node analysis diagram, one might be able to make a guestimate for the most costly exploit. One would be correct in assuming CVE-2012-1889 would fetch the highest value on the underground black market. CVE-2012-1889 exploited vulnerability in the Microsoft XML Core Services for the entire Microsoft Windows product line. In contrast, CVE-2010-2884 narrowly exploited a single vulnerability in the add-on Adobe Flash Player. While CVE-2012-1889 could be used universally against all Windows Operating Systems, CVE-2010-2884 required the victim to have specific software installed. Thus, CVE-2012-1889 would fetch a significantly higher value in the underground exploit market. Both exploits targeted visitors to the Hong Kong Amnesty International Page. So we must conclude that the information value of the targets would be the same. Thus, to clear the bar with the minimum cost for the attack, the

attackers reused encoders, payloads, packers, drop sites, and command and control servers from other exploits. This decision proved not to be cost-effective, as once 2012-0779 and 2012-1875 were identified in the wild, researchers identified CVE-2012-1889 through signature-based detection. This example provides us promise about a strategy to defend against attacks. The Elderwood Gang incorrectly assumed that their lower-bar attacks would go undetected and reused pieces for more costly attacks. Thus, to expose their subset of attacks, we only need to identify the lower-bar attacks and tie their pieces to the more costly attacks. Let's discuss this concept in more detail.

Clearing the Minimum Bar for Successful Exploitation

> *The chief assets of the attacker are…the power to choose at what place, by what method, and at what time the main action will be fought.*
>
> Field Marshal A. Wavell, *The Good Soldier*

In January 2010, Google revealed that they had been targeted in a large-scale attack alongside 20 other tech companies, including Yahoo, Symantec, Northrop Grumman, Morgan Stanley, and Dow Chemical (Binde, 2011). By exploiting and compromising employees in these companies, the Elderwood Gang pivoted and stole the intellectual property stored on each company's Software Configuration Management (SCM) servers. Dubbed Operation Aurora, the attack proved extremely successful in stealing millions (if not billions) of dollars of intellectual property. Yet, the media reports calling the attack advanced and sophisticated prove untrue after unraveling some of the technical details of the attack. Consider the temporaneous work of Peter Vreugdenhil, who won the 2010 Pwn2Own competition for developing a 0-day to exploit Internet Explorer. Figure 5 shows that his exploit earned him a mere $10,000 in prize winnings, yet it worked on the most current Operating System and software, bypassed two of the more complicated mitigation strategies, and chained multiple vulnerabilities to succeed. In

comparison, the Chinese-PLA sponsored Elderwood Gang utilized an exploit that affected a legacy browser, legacy operating system, and failed to bypass modern mitigation strategies.

	Pwn2Own 2010 Exploit	Operation Aurora Exploit
Affected Software	Internet Explorer 8	Internet Explorer 6
Affected OS	Windows 7 (released in 2009)	Windows XP (released in 2001)
Bypasses DEP	Yes	No
Bypasses ASLR	Yes	No
Chains Multiple Vulnerabilities	Yes	No
Development Team	1 Contestant	Nation-State Sponsored Team
Result	$10,000 in Prize Money	Intellectual property of 20 top tech companies.

Figure 5: Comparison of Winning 2010 Pwn2Own Exploit Vs. Operation Aurora Exploit

Based on their ultimate success, we can draw possible conclusions about our adversary's skillset in exploitation. First, we might falsely assume that the attackers lacked sophistication. The attackers used an exploit that targeted legacy systems and failed to bypass several of the more current mitigation strategies. This evidence would falsely support that conclusion. However, consider the attacker's strategy. They just *barely cleared the bar-to-entry to exploit twenty of the top tech companies* and successfully steal their intellectual property. Utilizing the minimal exploit necessary teaches us something about our adversary. *They are disciplined*. They are disciplined beyond belief in this domain of warfare. They understand military concepts like conservation of force and correctly apply these concepts to computer attacks. While the US Government invested $2.3 billion in 2012 to improve cyber[4] capabilities, our adversary invested

[4] The use of word cyber in this sentence is effectively the only use of the word in the entire chapter. (Cyber is used only two other times in a specific title for an exercise.) Here, cyber is used in a sentence that conveys our disproportional investment into a flawed defensive strategy.

a few thousand dollars in research and development costs to make a legacy exploit. Did the attackers have exploits that would succeed against modern Windows 7 and Internet Explorer 8? Definitely, as we saw in later attacks by the Elderwood Gang. Did they utilize those exploits during the Aurora campaign? No. Because they didn't need to in order to succeed. They simply needed to clear the minimum bar, get into the companies and steal the intellectual property before eventual discovery. This is the often misunderstood hidden side of asymmetric warfare: Spend the least amount necessary to cause the maximum impact. The attackers knew they would eventually be discovered. They knew their exploit would be burned. Instead of burning a more costly exploit, they burned a legacy exploit while still succeeding in their mission.

Fast-forward three years and our adversaries continue to learn and grow in their capabilities and understanding of warfare. Consider the attack we discussed earlier, where Chinese hackers compromised the Department of Labor Website in order to attack Department of Energy nuclear scientists who connected to the website (Ghosh, 2013). The attack included a novel vulnerability in Internet Explorer 8 (Lemos, 2013). Arguably, the attackers used a costly, more modern exploit; however, they proved their real talent only after successfully compromising the victims. Immediately after compromise, the attackers executed code to run extensive checks on the software versions of Java, Microsoft Office, Adobe Reader, Adobe Flash, antivirus, and browser plugins running on the system (Blasco, 2013). After investing heavily in a modern and costly 0-day to gain access, the attackers may have realized they would need a more inexpensive attack strategy for future attacks. Thus, the post-compromise activities may have been intended to find a vulnerability vector to gain the next access. After identifying the Java software running on a system, the attackers may invest in developing a less costly Java

exploit for future attacks.[5] Next, the attackers may test their malicious payloads to ensure they bypass the specific antivirus found on the system. Ultimately, post-compromise software checks can tell us two things about the attacker. Employing these tactics reminds us that the adversary is disciplined. However, what scares us more is that our attackers are in this for the long haul. They are dedicated to ensuring future attacks succeed just as much as current and previous attacks. Discussing previous attacks, its important to examine where our previous defenses have failed.

Failing to Understand Offense, We Fail to Defend

He who wants to protect everything, protects nothing.

General Adolf Galland, Luftwaffe

Let's examine some high-profile attacks that have succeeded over the last three years. Often the media and even well-respected security analysts quickly dub these attacks as sophisticated or complex. No doubt these attacks achieve epic results, including the theft of defense industrial plans, disruption of rebel networks, or the denial of service to banks. However, the exploits used in these attacks lack sophistication. It's not a method of using the simplest technology, as some would argue; in the advanced world of computer attacks, exploits that lack sophistication are simply easier to defend against. So why do these exploits succeed and our defenses fail? We will examine three specific attacks: the 2011 breach of RSA, the 2012 use of the Dark Comet RAT against Syrian rebels, and the 2013 Dark Seoul attack against the South Korean banking industry. All of these attacks included some form of malware and a vector to inject that malware. Yet these vectors lacked any real complexity or sophistication.

[5] In June 2013 alone, attackers wrote two separate Java remote code execution exploits for public release into the Metasploit Framework.

To understand a complex exploit, take a look at the result of Google's 2012 Pwnium Competition in Figure 6. An anonymous teenage hacker, known only as Pinkie Pie, wrote the winning exploit for the competition to exploit Google's Chrome browser. Built around a security framework, including an application sandbox, the Chrome browser contains a deep security framework. In fact, Pinkie Pie was the only hacker in the competition who was capable of writing an exploit for Google Chrome browser. In order to succeed, he chained six separate novel vulnerabilities together (Obes, 2012). The application succeeded in bypassing all modern exploit mitigation techniques and earned the hacker $60,000 in prize earnings (Obes, 2012).

	# of Chained Vulnerabilities	Bypasses Application Sandbox	Bypasses ASLR	Bypasses DEP	Preventable	Result
2012 Pwnium Competition (Google)	6	Yes	Yes	Yes	No	$60,000 prize winnings
RSA Breach (US)	1	No	No	No	Yes	Stole F-35 Strike Force Fighter Plans
Dark Comet (Syria)	0	No	No	No	Yes	Disrupted Syrian Rebel Network
Dark Seoul (South Korea)	1	No	No	No	Yes	Disrupted South Korean Banking

Figure 6: Comparison of 2012 Pwnium Exploit Competition with High-Profile Attacks

Compare Pinkie Pie's exploit to the 2011 breach of the RSA Networks. In an effort to target victims in the defense industrial base, Chinese hackers targeted members of the RSA Company. RSA, a security company, builds secure ID tokens that grant limited, two-factor authentication for additional security. Realizing that the tokens were in widespread use in the defense industrial base, the hackers emailed a spreadsheet to employees at RSA (O'Connor, 2013). The malicious spreadsheet exploited a vulnerability in handling Flash. Once they had compromised the employees, the hackers stole the proprietary source code from RSA. After examining the source code of RSA Secure ID, the hackers used the knowledge gained in separate attacks against Lockheed Martin and other companies. The exploit used in the attack was nowhere near the same level of sophistication as Pinkie Pie's winning exploit. For example, if the defenders at RSA had enabled Data Execution Protection (a defense mitigation strategy available for six years), the exploit would have failed. If the defenders at RSA installed and properly configured Microsoft Enhanced Mitigation Experience Toolkit (a free Microsoft Security Tool for two years), the exploit would have failed (O'Connor, 2013). If the defenders simply segregated their network and didn't allow the entire company unfiltered access to the source code repositories, the exploit would have succeeded but the attackers would have been able to steal the source code. For a company that sells security solutions, the defense team at RSA was sloppy.

Let's discuss a separate attack that targeted dissidents in Syria in 2012. Dedicated to gaining intelligence about the dissidents, the attackers distributed malware disguised as a fake Skype encryption tool (Fisher, 2012). Once installed, the malware logged keystrokes, captured webcam images, installed a module to maintain persistence, and conducted a range of surreptitious activities (Fisher, 2012). The malware contained a common malicious remote

access toolkit known as Dark Comet. Available for download for a few years, the malware proved nothing extremely sophisticated. Yet when the Electronic Freedom Foundation (EFF) discovered the attack in August 2012, most popular antivirus software versions could not detect the malware using signature-based detection (Fisher, 2012). Briefly consider the implications: this particular malware existed for several years on a public website and antivirus vendors did not write a signature for it. Understanding this, we begin to realize the lack of usefulness in most antivirus vendor products. Instead of examining the indicators of malicious activity (for example, logging keystrokes, turning on the webcam, installing a persistence module), most antivirus programs use a sequence of bytes or an MD5 hash to identify a malicious program. While modern enterprise networks and home users cite antivirus programs as one of their primary methods for defense, you can begin to see the futility of it when considering the Dark Comet attack in 2012.

In 2013, an attack against the South Korean banking industry crippled the nation's resources for a day. In March 2013, attackers disabled thousands of computers in a coordinated attack (O'Connor, 2013). While the initial reports sound sophisticated, pause and examine exactly what happened. The initial vector is unknown, but most likely the attackers exploited victims using a watering-hole or drive-by-download technique, which downloaded malware onto the victims' computers. Upon gaining access to the victim's machines, the attacker's malware killed two antivirus processes and then wrote over the master boot record (MBR) of every hard drive on the victims' machines. Writing over the MBR of a drive ensures it cannot boot, leaving the machine in an inoperable state, but low-level commands like writing the MBR require administrator-level permissions. How did the attackers change their permissions from a victim user to an administrator so quickly? You might conclude that they developed a costly privilege

escalation technique. Nope. You might offer that they compromised administrator passwords in advance of the attack. Nope. You might think they found some way to bypass privilege checks and write over the MBR as an unprivileged user. Nope. What happened? The victim users (at the banks) were running as administrators. They failed to separate user-level permissions and administrator permissions. So any user, including those compromised, had the ability (on a bank's computer) to run low-level commands. This is the problem with our current defense posture: We have made the bar so low that an attacker has to barely invest to win. If the attackers had to invest in costly privilege escalation techniques, they may have failed or chosen a different course of action, but because the investment to attack was so low, the attackers were able to wreak havoc on thousands of machines with very little investment. Understanding this, we must propose a new way to defend systems.

Proposing an Asymmetric Defense Strategy

War is the unfolding of miscalculations.

Barbara Tuchman

While our defense strategy over the last decade may appear hopeless, all is not lost. Embracing a little bit of knowledge, we can prepare asymmetric defenses. With knowledge of an attacker's strategy, we can invest in minor defenses that will cause a dramatic and costly shift for the attacker. Let's address three specific examples of how creativity is changing the balance of symmetry on the battlefield today.

How Undergraduate Students Beat the NSA Red Team

> *In the development of air power, one has to look ahead and not backward and figure out what is going to happen, not too much what has happened.*
>
> Brigadier General William "Billy" Mitchell, *Winged Defense* (1925)

Consider the strategy employed by the 2011 National Cyber Defense Exercise Champions from the US Military Academy. The Cyber Defense Exercise (CDX) is a National Security Agency-sponsored exercise where undergraduate students are forced to defend a series of unpatched machines (without antivirus defenses) from attack by the elite NSA red team. The premise seems unfair from the onset. How do you defend vulnerable machines from attack by a determined and skilled adversary? *To succeed you must think like the enemy*. You must become an attacker if you stand any chance of laying a formidable defense.

To understand this, let's examine one of the many attack vectors the students successfully defended against. Specifically, let's examine the possibility that an attacker may attempt to use a specially crafted malicious Adobe Portable Document Format (PDF) file to gain remote access to a victim. During the exercise, the NSA Red Team used open-source attack frameworks. Arguably, Metasploit is the most popular of these exploit development frameworks. Developed by the legendary hacker H. D. Moore, Metasploit contains well over a thousand unique exploits and a framework to build many more. Twelve of these exploits specifically target Adobe Acrobat software. Browsing Figure 7, one might notice an interesting aspect of these exploits: Eleven of the twelve PDF exploits require the use of JavaScript and Automatic Action.[6]

These PDF document reader exploits specifically require JavaScript to run as soon as the document is opened in order to place malicious code into Adobe Acrobat's process heap.

---

[6] The twelfth malicious PDF module relies more on a social-engineering trick and less on a technical flaw in the software.

Metasploit Exploit Module	Affected Version	CVE	Date Released	Utilizes JS & Auto Action
adobe_collectemailinfo	8.1.1	2007-5659	8-Feb-08	Yes
adobe_utilprintf	8.1.3	2008-2992	8-Feb-08	Yes
adobe_jbig2decode	9.0.0	2009-0658	19-Feb-09	Yes
adobe_geticon	9.1	2009-0927	24-Mar-09	Yes
adobe_flatedecode_predictor	9.2	2009-3459	8-Oct-09	Yes
adobe_u3d_meshdecl	9.3	2009-3953	13-Oct-09	Yes
adobe_media_newplayer	9.2	2009-4324	14-Dec-09	Yes
adobe_pdf_embedded_exe	9.3.3	2010-1240	29-Mar-10	Yes
adobe_pdf_embedded_exe_nojs	9.3.3	2010-1240	29-Mar-10	No
adobe_cooltype_sing	9.3.4	2010-2883	7-Sep-10	Yes
adobe_libtiff	9.3	2010-0188	16-Dec-10	Yes
adobe_reader_u3d	10.1.1	2011-2462	6-Dec-11	Yes

Figure 7: Metasploit Adobe Reader Modules

Without spraying the heap, the malicious PDF documents cannot execute further malicious instructions and thereby fail to exploit their targets. Analyzing a larger sample, we downloaded 9,000 clean and 9,000 malicious PDF documents from the Contagio Malware Repository (O'Connor, 2013). Of the 9,000 malicious PDF documents, 92% specifically used JavaScript and Automatic Action. In stark contrast, only 2% of the clean PDF documents required JavaScript and Automatic Action (O'Connor, 2013). Thus, removing and flattening the JavaScript code will ensure that the document fails to perform its malicious purpose. After a minor investment in studying PDF malware, suddenly we can prepare an adequate PDF malware defense that will reduce the threat landscape by over 90%. This is the role of an asymmetric defense—small investment, big yield.

West Point Cadets Anthony Rodriguez and Robert Frost prepared this defense by writing software that scrubbed PDFs entering the West Point network perimeter. With a nod to offense, the software they wrote utilized Python PDF parsing libraries from Didier Stevens (the author or co-author of all twelve of the Metasploit PDF modules). The result? The NSA Red Team aggressively tried to compromise machines using PDF malware. Attempt after attempt, they failed. In fact, at one point the victim terminals had to be examined to ensure they were not running antivirus software and that they had the correct (vulnerable) version of Adobe Acrobat. When given the option to patch the machines with updated software later in the exercise, the two cadets declined, mocking their attackers. The attackers, convinced that their PDF exploits should have worked, disproportionally invested in a flawed attack strategy.

Three Hundred Lines of Python Wrapping Around the Department of Energy

> *Air power can either paralyze the enemy's military action or compel him to devote to the defense of his bases and communications a share of his straitened resources far greater that what we need in the attack.*
>
> Sir Winston Churchill, UK Prime Minister

Is there is a huge difference, though, between a five-day undergraduate exercise and the attacks in the real world? Not really. Consider the recent work by Matt Myrick at the Department of Energy's Sandia Labs. Frustrated by a small federal budget and engaged in a constant battle with advanced threats, Myrick developed a tool to block malicious websites, hashes, and spear-phishing-attacks. His tool, Master Block List, contains a mere 300 lines of code that runs at the perimeter of the network (Higgins, 2012). Admins at Sandia Labs, Los Alamos Labs, and DOE's Pantex Plant have embraced the toolkit, sharing indicators of attack amidst the three plants (Higgins, 2012). Myrick's successful efforts highlight where we are failing in the domain of cyber defense: *we lack creativity.* Furthermore, we'd argue that we not only lack creativity but

also *reject and deter creativity* in defense. *Creativity lacks a formal process.* Think for a second about Matt's toolkit. Does it have a DoD Certificate of Networthiness? No. Has it been tested in an NSA approved network lab? Certainly not. In fact, how much time did Myrick spend writing his tool? Probably less than twenty hours in total. But Matt *wisely invested twenty hours of creativity*. As a result, he is single-handedly changing the tactics, techniques, and procedures for how the adversary attacks. Will his toolkit completely stop an advanced adversary? No. But now the adversary has to custom design malware for Sandia, separate malware for Los Alamos, and separate malware for Pantex. If Myrick's tool sniffs a hint of malicious activity at one plant, it immediately shares the information with the other plants, ceasing traffic and stopping the attack in real time. This is the beauty of asymmetry. Myrick, as an individual, can make the cost of stealing information from Sandia more expensive than the adversary can afford to pay. At this point, Myrick wins.

What Happens When You Hire the Best Hackers

Nothing can stop the attack of [hackers] except other [hackers].

Major General William "Billy" Mitchell, *Winged Defense* (1925),
updated by Matt and Kirk, US Army Special Operations (2013)

We have made the argument that creativity is lacking in defense, but attackers have an abundance of creativity. Consider the concept of Return-Oriented Programming, or ROP. After developing Data Execution Protection (DEP), Microsoft made it rather difficult for attackers to write exploits using the standard buffer overflow technique. Essentially, attackers lost the ability to place their malicious code onto the program's stack and execute it. Hardware-enforced DEP makes the program's stack non-executable. To circumvent this strategy, hackers developed the technique of ROP Gadgets. Instead of placing executable code on the stack, they place a series of borrowed addresses on the stack which point to a small set of instructions. Each one of these

gadgets executes a small bit of the overall malicious program and then returns control to the stack. Some of the most brilliant minds in information security worked on this problem. Matt Miller was one of these hackers. Additionally, Matt was the third developer at the open-source Metasploit Framework. Miller is one of the most creative and divergent thinkers in the field of computer security.

So when Microsoft set out to develop a more intelligent security design for Windows 8, they first consulted with and then hired Miller. One of many on the very creative Microsoft Security Engineering Team, Miller worked on several of the exploit mitigation strategies for Windows 8. What happens when you try to use ROP Gadgets on Windows 8? The program checks who you are and if you should be logically calling those addresses. If you don't pass the common-sense test, the program terminates. Now, the strategy is not flawless, but right out of the box, Windows 8 reduces the threat landscape by stopping 85% of existing malware. Miller's team dedicated themselves to creativity in defense.

In their constant search for creative means of defense, Microsoft held the 2012 Blue Hat competition. A single Ph.D., Ivan Fratric, won the competition with a novel means for further detecting ROP gadgets in use (Microsoft, 2012). After giving Fratric a $200,000 prize, the Microsoft team integrated his novel defense means into their Enhance Mitigation Experience Toolkit (EMET). Compare the $200,000 investment against the value of the information stolen in the multitude of attacks against businesses running Windows. Two hundred thousand dollars is a minor investment. It took the Microsoft team only three months to add Fratric's proof of concept to EMET (Microsoft, 2012). Three months, $200,000, one Ph.D. candidate. This is the beauty of an asymmetric defense. A single individual, a single investment, a single effort can disproportionally affect the adversary and force a change in adversarial tactics.

Concluding Thoughts

> *The most important branch of aviation is pursuit, which fights for and gains control of the air.*
>
> Brigadier General William "Billy" Mitchell

In the previous chapter, we discussed some thoughts behind the current state of attacks and defense against our computer networks. We began by examining the changing adversarial tactics. Dissecting a watering-hole attack against the Department of Energy, we concluded our adversaries are disciplined in conservation of force. Like the adversary, we realized the importance of utilizing the absolutely least costly method. Examining the specific costs of an exploit, we focused on the hidden cost of attribution. Studying a series of attacks by the Chinese Elderwood Gang, we examined how attackers accidentally reveal their capabilities by failing to recognize this hidden cost of attribution.

We challenged the myth that the current adversary has used advanced and unstoppable tactics to compromise computer systems. In our examination, we compared successful attacks to contemporary exploit development competitions of Pwn2Own and Pwnium. To further illustrate that current attacks are unsophisticated, we discussed the 2011 RSA breach, the 2012 Dark Comet, and 2013 Dark Seoul attacks. All three attacks helped us understand that the adversary is just barely clearing the bar with the least costly technology. To cease future attacks, we proposed an emphasis on asymmetry in future defense strategies. Instead of an investment into billions of architecture, we proposed a more valid investment into creativity. We highlighted the 2011 Cyber Defense Exercise champion's strategy to beat the NSA, impromptu code written by an employee at the Department of Energy, and Microsoft's efforts to hire one of the best hackers as asymmetric defense strategies.

In conclusion, we are at the heart of this fight. We, much like our adversaries, are defining the tactics, techniques, and procedures for how we will fight for generations to come. *If you leave this chapter with one thought*, ***understand that holistic and permanent computer network defense is impossible despite massive resource investment***. Yet, *we can always win by forcing the adversary to spend more on the attack than the value of the information stolen or the systems broken*. We can always win when individuals are capable of changing the tactics of an entire adversary. However, we will only achieve these asymmetric results when *we embrace creativity on the battlefield.*

References

Binde, B., McRee, R., & O'Connor, T. J. (2011, May 1). Assessing outbound traffic to uncover advanced persistent threat. *SANS Technology Institute*. Retrieved June 23, 2013, from www.sans.edu/student-files/projects/JWP-Binde-McRee-OConnor.pdf

Blasco, J. (2013, May 1). U.S. Department of Labor website hacked and redirecting to malicious code. *AlienVault Labs*. Retrieved June 12, 2013, from http://labs.alienvault.com/labs/index.php/2013/u-s-department-of-labor-website-hacked-and-redirecting-to-malicious-code/

Doyle, D. (2001, August 1). Code red: The one to not "dew". *SANS Reading Room*. Retrieved June 30, 2013, from www.sans.org/reading_room/whitepapers/malicious/code-red-dew_66

Eeckhoute, P. V. (2013, February 19). DEPS—Precise heap spray on Firefox and IE10. *Corelan Team*. Retrieved June 11, 2013, from https://www.corelan.be/index.php/2013/02/19/deps-precise-heap-spray-on-firefox-and-ie10/

Fisher, D. (2012, August 16). DarkComet RAT used in new attack on Syrian activists. *Threatpost*. Retrieved June 23, 2013, from http://threatpost.com/darkcomet-rat-used-new-attack-syrian-activists-081612/

Ghosh, A. (2013, May 1). Part 1—K.I.A.—U.S. Dept. Labor website pushing Poison Ivy—CVE-2012-4792 | Invincea. *Invincea*. Retrieved June 16, 2013, from http://www.invincea.com/2013/05/k-i-a-us-dol-website-pushing-poison-ivy-cve-2012-4792/

Halfacree, G. (2012, November 2). VUPEN sells Windows 8 zero-day vulnerability code | bit-tech.net. *bit-tech.net*. Retrieved June 16, 2013, from http://www.bit-tech.net/news/bits/2012/11/02/win8-zero-day/1

Higgins, K. J. (2012, October 3). Government agencies get creative In APT battle. *Dark Reading*. Retrieved June 23, 2013, from http://www.darkreading.com/government-vertical/government-agencies-get-creative-in-apt/240008438

Johnson, K., & Miller, M. (2012, July 31). Exploit mitigation improvements in Windows 8. Blackhat USA 2012 Conference, Las Vegas, NV.

Krebs, B. (2013, January 16). New Java exploit fetches $5,000 per buyer. *Krebs on Security*. Retrieved June 16, 2013, from krebsonsecurity.com/2013/01/new-java-exploit-fetches-5000-per-buyer/

Lemos, R. (2013, May 4). Zero-day exploit enabled cyber-attack on U.S. Labor Department. *eWeek.com*. Retrieved June 16, 2013, from http://www.eweek.com/security/zero-day-exploit-enabled-cyber-attack-on-us-labor-department/

Meer, H. (2010, June 25). Memory corruption attacks the (almost) complete history. *Black Hat Briefings 2010*. Retrieved June 28, 2013, from http://media.blackhat.com/bh-us-10/whitepapers/Meer/BlackHat-USA-2010-Meer-History-of-Memory-Corruption-Attacks-wp.pdf

Microsoft security toolkit delivers new BlueHat prize defensive technology. (2012, July 25). *Microsoft Corporation*. Retrieved June 29, 2013, from http://www.microsoft.com/en-us/news/Press/2012/Jul12/07-25BlueHatPrizePR.aspx

Microsoft security advisory (2847140): Vulnerability in Internet Explorer could allow remote code execution. (2013, May 14). *TechNet*. Retrieved June 16, 2013, from http://technet.microsoft.com/en-us/security/advisory/2847140#section1

Miller, C. (2007). The legitimate vulnerability market: Inside the secretive world of 0-day exploit sales. Paper presented at the 2007 Workshop on the Economics of Information Security, June 7–8, 2007, Pittsburgh, PA. Retrieved June 11, 2013, from http://weis2007.econinfosec.org/papers/29.pdf

Mitchell, W. (2009). *Winged defense the development and possibilities of modern air power—Economic and military*. Tuscaloosa, AL: University of Alabama Press.

Obes, J. L., & Schuh, J. (2012, May 22). Chromium blog: A tale of two Pwnies (Part 1). *Chromium Blog*. Retrieved June 23, 2013, from http://blog.chromium.org/2012/05/tale-of-two-pwnies-part-1.html

O'Gorman, G., & McDonald, G. (2012, September 16). The Elderwood Project. *Symantec*. Retrieved June 16, 2013, from http://www.symantec.com/connect/blogs/elderwood-project

O'Connor, T. J. (2013, January 2). EMET 3.5: The value of looking through an attacker's eyes. *SANS Internet Storm Center*. Retrieved June 13, 2013, from https://isc.sans.edu/diary/EMET+3.5%3A+The+Value+of+Looking+Through+an+Attacker's+Eyes/14797

O'Connor, T. J. (2013, April 18). Don't pull your goalie: The role of active defense. *ForenSecure 2013*. Lecture given at the Illinois Institute of Technology, Wheaton, IL.

Pwn2Own 2013 Contest Rules. (2013, March 5). *TippingPoint | DVLabs*. Retrieved June 16, 2013, from http://dvlabs.tippingpoint.com/Pwn2OwnContestRules.html

Sinn3r. (2013, March 4). New heap spray technique for Metasploit browser exploitation. *Security Street Rapid7*. Retrieved June 11, 2013, from https://community.rapid7.com/community/metasploit/blog/2013/03/04/new-heap-spray-technique-for-metasploit-browser-exploitation

Sinn3r. (2013, May 5). Department of Labor IE 0-day exploit now available at Metasploit. *Security Street Rapid7*. Retrieved June 11, 2013, from https://community.rapid7.com/

community/metasploit/blog/2013/05/05/department-of-labor-ie-0day-now-available-at-metasploit