LESSONS LEARNED: CAN ALERTING THE PUBLIC ABOUT EXPLOITATION DO MORE HARM THAN GOOD?

Holly Stewart
Microsoft, Redmond, Washington, USA
Email hollyst@microsoft.com

Tom Cross
Lancope Inc., Alpharetta, Georgia, USA
Email tcross@lancope.com

ABSTRACT

Much has been written about the ethics and timing of security vulnerability disclosure. But what about the ethics and timing of public disclosure of the fact that a vulnerability is being exploited in the wild? Information about in-the-wild exploitation is important for prioritizing defensive efforts. However, knowledge that a vulnerability exists and can be targeted effectively in practice can attract the attention of attackers and accelerate the amount of attack activity taking place. This paper reviews numerous cases that span many years of active exploitation data to evaluate the real-world consequences of exploitation disclosure. These examples help illustrate when exploitation information can best aid the public in defending itself against attacks and when it actually encourages more attacks. Based on these examples, we provide actionable guidance about the timing and coordination of exploitation disclosure that can be utilized by anyone who might be involved in this process, from vulnerability researchers to the targets of exploitation, the media and even the vendors themselves.

1. INTRODUCTION

Many significant security vulnerabilities have been discovered over the years by independent security researchers. A great deal of consideration has been given to the ethical questions facing such an independent researcher regarding whether to disclose those vulnerabilities publicly or to privately inform the responsible software vendors. Of course, some independent security researchers aren't interested in disclosing vulnerabilities at all – they seek to use them to launch attacks.

When vulnerabilities are privately discovered and then used in attacks, they may be uncovered by security professionals in the course of analysing new malware samples or investigating breaches. These security professionals are faced with a slightly different ethical dilemma from that faced by an independent security researcher. Not only is the vulnerability in question potentially unknown to the responsible software vendor, but there is attack activity going on that needs to be stopped as soon as possible. As these circumstances have become increasingly common, it is important to understand the associated ethical considerations.

Should security professionals inform the public when they discover that a new vulnerability is being targeted in the wild? When and under what circumstances?

It can be important to disseminate the knowledge that a vulnerability is being targeted by attackers quickly. Software vendors and IT professionals need to understand how to prioritize vulnerability remediation, and the fact that exploitation is occurring can motivate faster release and deployment of the remediation. Security product vendors need access to real-world exploit samples so they can validate coverage. Network managers need to know in real time what attacks are taking place, so they can be prepared and focus their attention on the right warning signs and mitigations. End-users also need to know what the overall threat environment is on the Internet.

However, the timing of and details related to exploitation disclosure can also escalate the general use of a new exploit. Thousands [1] of new software security vulnerabilities are disclosed publicly every year. Some are much more suitable for launching attacks than others. Attackers are attracted to vulnerabilities that have successfully been used by other attackers. The knowledge that a particular vulnerability exists and has been targeted 'in the field' can indicate to attackers that it is worth their time and effort to investigate that vulnerability and reproduce a functional exploit or integrate a public one into their toolkit.

This paper reviews examples of real-world disclosure of exploitation that occurred at various stages within the vulnerability disclosure process. Our examples speak to the many variables associated with live exploitation, from small-scale targeted attacks to large-scale, malicious toolkit integrations that reach tens of thousands of people.

These examples illustrate when exploitation information can best aid the public in defending itself against attacks by prioritizing remediation efforts, and when it actually encourages more attack activity. With these examples in mind, we provide actionable guidance about the timing and coordination of exploitation disclosure that can be utilized by anyone who might be involved in this process, from vulnerability researchers to the targets of exploitation, the media and even the vendors themselves.

Section 2 of this work provides a background on security vulnerability disclosure ethics. Section 3 introduces the concept of exploitation disclosure and provides some real-world examples that illustrate its impact. Section 4 considers the consequences of exploitation disclosure in light of the amount of attack activity occurring at a particular time and in light of the amount of information that is available about the vulnerability at a particular time. Section 5 puts these ingredients together, providing guidance on the timing of coordinated exploitation disclosure in light of all of these factors. Section 6 presents our conclusions.

2. BACKGROUND ON VULNERABILITY DISCLOSURE ETHICS

Debate about the ethics of security vulnerability disclosure by independent security researchers has traditionally occurred between two camps: the full disclosure camp and the coordinated disclosure camp.

The full disclosure camp argues that informing the public about a security vulnerability allows end-users and administrators to understand exactly what is wrong with the security of their systems and take immediate action to protect themselves from attacks. This camp also maintains that full

disclosure forces software vendors to quickly provide their customers with updates or workarounds to address the vulnerability, rather than ignoring the issue or dismissing it as hypothetical [2].

The coordinated disclosure camp argues that full disclosure provides information that can be useful to attackers and can lead to an increase in attack activity [3]. This camp argues that researchers should inform software vendors privately about security vulnerabilities and wait until updates are available before informing the public about the vulnerability [4]. This approach is sometimes referred to as 'responsible disclosure' [5].

Over time, norms of behaviour have emerged in which different actors balance these two approaches. Many independent security researchers engage in coordinated disclosure, but the threat of full disclosure remains as a forcing function to make vendors address vulnerabilities rapidly [2]. Detailed information and exploit code often emerges for vulnerabilities after public disclosure. This information allows the IT community to understand the underlying technical issues involved with the vulnerability, without providing attackers with exploit code before updates are available. Improvements to the speed at which computer networks deploy updates for security vulnerabilities have also helped limit the negative impact of public disclosure of detailed technical information after updates are released.

One of the key assumptions associated with the argument that it is best to privately coordinate the disclosure of an independently discovered security vulnerability is the assumption that it is unlikely that a malicious third party will discover the exact same vulnerability and begin to target it in the wild while the vendor is still working on an update. Without access to a large-scale system for monitoring exploitation, it might be a challenge for the finder to assess the validity of that assumption in the moment.

Public disclosure of a vulnerability has the effect of informing the bad guys about an opportunity that they can take advantage of. Therefore, public disclosure always has a negative consequence. The question of whether or not it is the right thing to do rests on whether or not it also has positive consequences, and ultimately whether or not those positive consequences outweigh the negative ones.

If a vendor is informed privately of a previously unknown vulnerability and manages to get their update out before malicious actors independently discover that vulnerability, then coordinated disclosure has been successful. With the benefit of hindsight, we can say that in such a case, public disclosure of the vulnerability before the update was available probably would have done more harm than good. There were no actual attacks that users and operators could have mitigated with the prior knowledge of the vulnerability. However, prior public disclosure would have armed attackers and led to attacks.

The longer it takes a software vendor to provide an update for a vulnerability, the greater the likelihood that a malicious actor will independently discover that vulnerability and begin to attack vulnerable systems. This clock is ticking during any vulnerability remediation process. Eventually, if it does not appear that the vendor will ever provide an update that addresses the vulnerability, many argue that the right thing to do is to disclose the vulnerability to the public, so that users of the software can take their own actions to mitigate the risk. This argument rests on the assertion that, given enough time, malicious actors will eventually discover the vulnerability even if it isn't disclosed.

But what if we are sure that the attackers already know about the vulnerability and they are already launching attacks that target it? Does it not make sense in this circumstance to inform the public about the vulnerability immediately, in order to level the playing field between the good guys and the bad guys on the Internet?

One reason for caution is that public disclosure may result in an increase in the amount of attack activity that is occurring. An empirical study performed by *Symantec Research Labs* [6] included data on the level of exploitation of several vulnerabilities both before and after public disclosure. This data demonstrates that public disclosure of vulnerabilities with pre-existing histories of attack activity has repeatedly coincided with substantial increases in that attack activity. A chart from that study visually communicates the story very well and is reproduced in Figure 1 below.

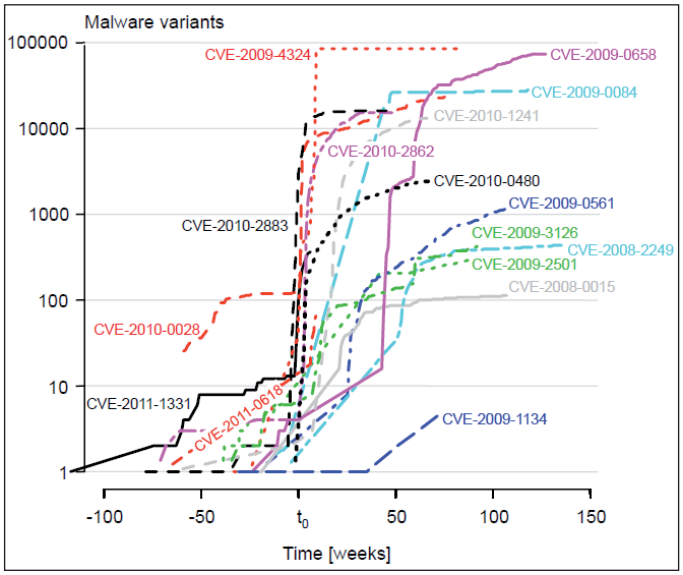

Figure 1: Malware variants [6].

Therefore, many security professionals find themselves faced with a challenging question when they discover a new vulnerability that is being targeted in the wild – will immediate disclosure be more harmful or helpful?

3. EXPLOITATION DISCLOSURE RISKS AND REWARDS

The first step to unravelling this ethical question is to recognize that the fact that a vulnerability is being exploited in the wild is a separate piece of information from the fact that the vulnerability exists in the first place, and it is important to ensure that both of these pieces of information are eventually publicly disclosed.

When we refer in this paper to information about exploitation, we are referring to high-level information about the fact that the vulnerability is being exploited in the wild, who is exploiting it, and how much attack activity is occurring. We are *not* referring to technical details about how to exploit the vulnerability. In our view, it is not ethical in most cases to

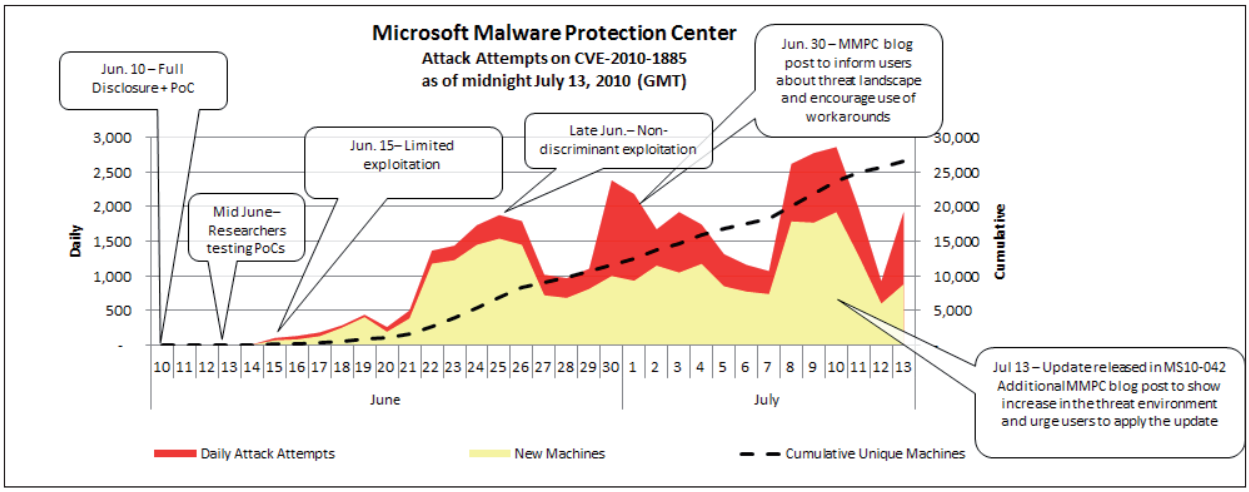

Figure 2: Attack attempts on CVE-2010-1885.

publicly disclose technical details about how to exploit a vulnerability until long after an update is available that addresses the vulnerability in question.

However, if you are aware that a vulnerability is being targeted in real-world attacks, the public needs to know, and in some cases it may be important to disclose that information to the public even before an update is released. Dissemination of information about exploitation can have a number of positive effects, which we will illustrate with three real-world examples. However, the timing is critical. Premature dissemination of information about the fact that exploitation is occurring in the wild can have negative consequences, which we will also illustrate with examples.

For each example, a timeline of attack activity is provided using information about attack detections that has been published by the *Microsoft Malware Protection Center* as well as *IBM X-Force Managed Security Services* [7]. These timelines are labelled with key disclosure and exploitation events.

A. Positive consequences of exploitation disclosure

Positive outcomes associated with exploitation disclosure include the following:

- **Mitigation prioritization:** If updates or workarounds are available, public knowledge of exploitation helps users and administrators prioritize the deployment of those mitigations.
- **Update prioritization:** Knowledge of the existence and prevalence of active exploitation helps the vendor of the affected software prioritize the release of an update that addresses the vulnerability.
- **Security product quality:** Real-world samples help vendors of third-party security products validate or improve the quality of the protection they have provided to their customers.

Mitigation prioritization

Our first example is related to a vulnerability disclosure that was not coordinated with the vendor of the affected software. In this case, the uncoordinated public disclosure of the vulnerability led to in-the-wild exploitation activity before an update was available. However, the vendor took steps to inform the public about the amount of exploitation activity that was occurring and provided mitigation advice. This example illustrates the fact that information about exploitation activity can help potential victims prioritize their efforts to mitigate a vulnerability when they have clear steps to follow.

CVE-2010-1885 was a vulnerability in *Microsoft Windows Help and Support Center* [8]. Because this was not a coordinated disclosure, an update was not available from *Microsoft* when the vulnerability and details about how to exploit it were publicly disclosed in early June 2010. However, when the vulnerability was disclosed, *Microsoft* did release an advisory that provided advice on temporary mitigation techniques and workarounds. It was about five days after disclosure before *Microsoft* detected in-the-wild attack activity targeting the vulnerability, although the activity was initially limited in scope. By late June, *Microsoft* was detecting attacks that were reaching many victims in a multitude of countries – the attacks were growing broader and less discriminant.

Over the course of June [9] and July [10], *Microsoft* published several blog posts about CVE-2010-1885 that included details about the amount of in-the-wild exploitation activity that it was seeing and which urged the public to implement the workarounds (Figure 2). These blog posts informed news media coverage of the issue and raised awareness about the importance of mitigating the vulnerability in light of the real-world threat environment.

Update prioritization

Our second example illustrates the fact that privately disclosed information about exploitation can help prioritize the efforts that software vendors are engaged in to develop and release updates for security vulnerabilities. A security professional named Mila Parkour received information that a vulnerability in *Adobe Flash Player* (CVE-2011-0611 [11]) was being exploited in targeted attacks. Ms. Parkour coordinated information about these attacks [12], including exploit samples, with *Adobe* as well as members of the

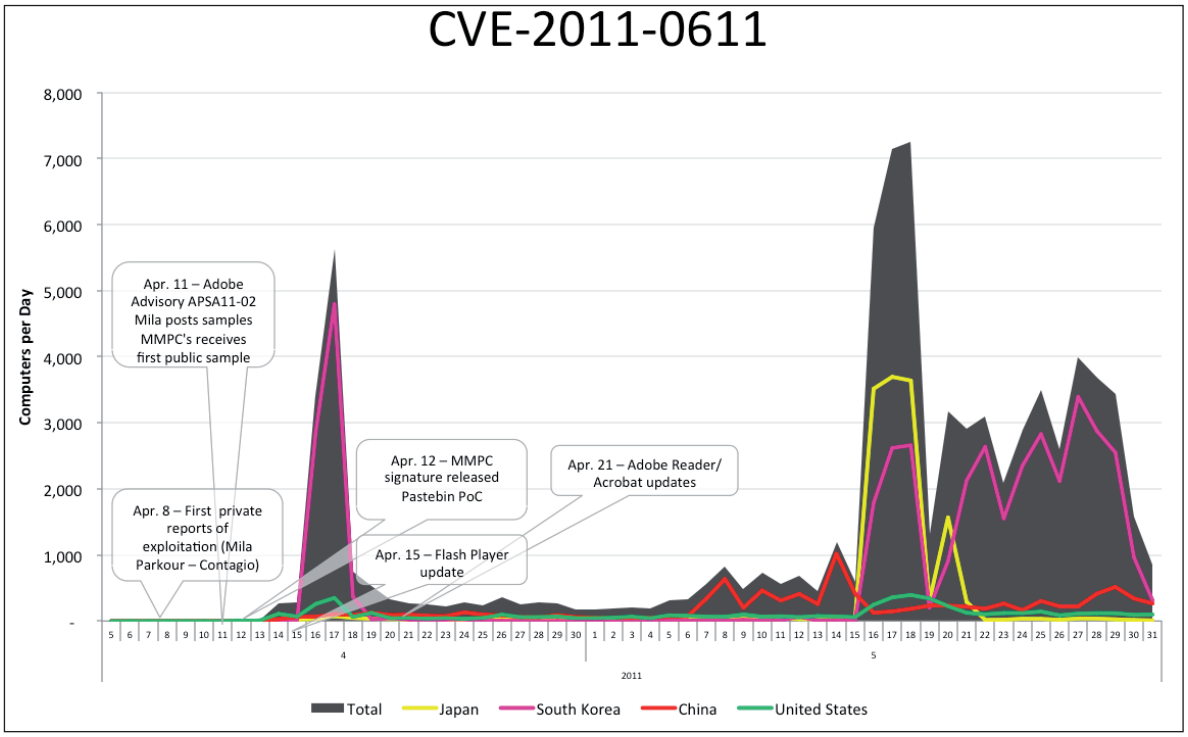

Figure 3: CVE-2011-0611 timeline.

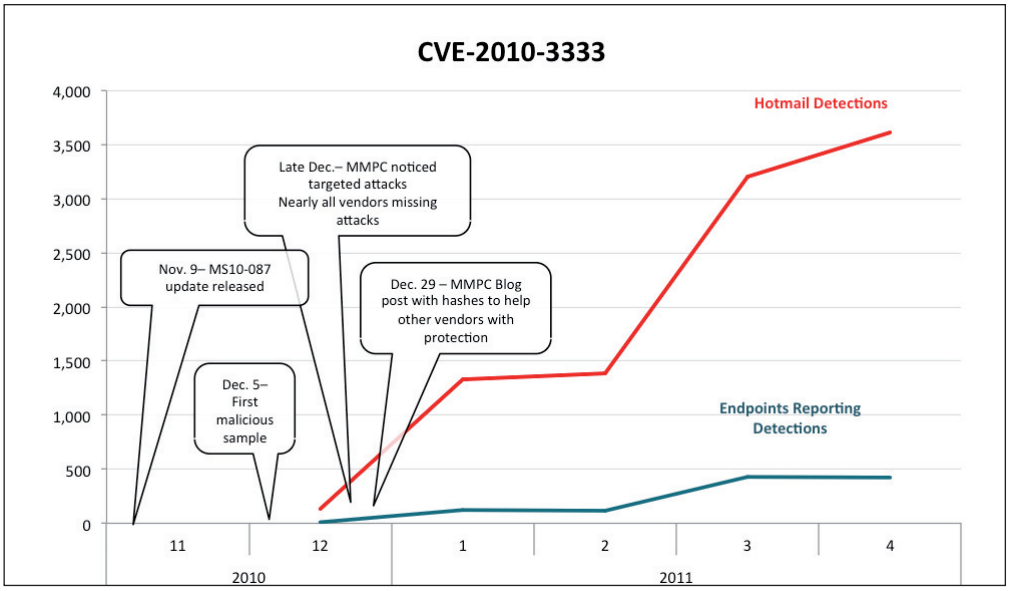

Figure 4: CVE-2010-3333.

security research community [13]. As a consequence of this information, *Adobe* had several days of lead time to prepare an update for the vulnerability before an exploit was publicly disclosed.

Attackers launched the first wave of widespread attacks targeting CVE-2011-0611 on the same day as the update for *Adobe Flash Player* was released. Additional waves of attacks happened in mid-May, long after updates had been made available (Figure 3). It's important to recognize that attacks move from targeted to widespread as information about vulnerabilities disseminates among attackers. If *Adobe* had not received word that exploitation was occurring when it did, it is possible that widespread attacks might have started before an update was available.

Security product quality

Our third example illustrates the fact that vendors of third-party security products need real-world exploit samples in order to verify that their products are working correctly. CVE-2010-3333 was a vulnerability in *Microsoft Word* that was disclosed to *Microsoft* by a third party [14]. The vulnerability was publicly disclosed along with the update on 9 November 2010. *Microsoft* also shared information about the vulnerability with vendors of computer security products

through the *Microsoft Active Protections Program* (*MAPP*).

The first real-world attack activity targeting the vulnerability began about a month after it was disclosed (Figure 4). At first, exploitation was targeted, but by the end of December, attacks had broadened. *Microsoft* noticed that several commercial security products were not detecting the exploit samples that were being disseminated in the wild, in spite of the fact that those products were supposed to detect and block attacks that targeted this vulnerability. *Microsoft* coordinated the sharing of samples with security software vendors so that they could test their products and update their signatures if necessary [15]. Even though an update was available for this vulnerability, information about in-the-wild exploitation was vital for protecting networks from attack.

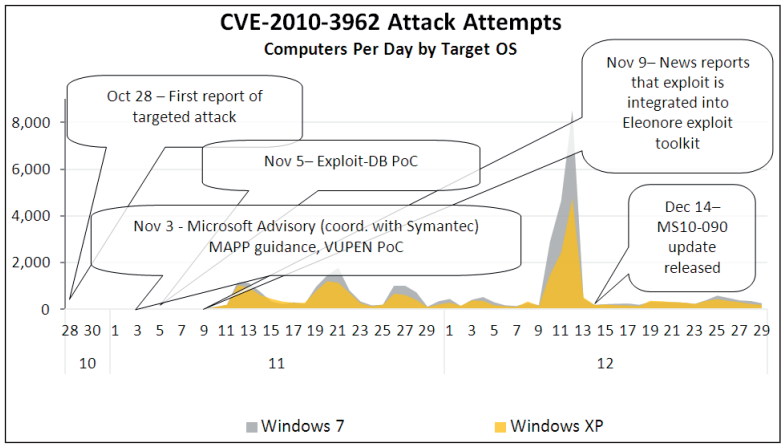

Figure 5: CVE-2010-3962 attack attempts.

B. Negative consequences of premature exploitation disclosure

We've established that exploitation disclosure has many virtues, but it is also important to understand that it can have negative consequences as well.

In October 2012, a zero-day vulnerability affecting *Internet Explorer* was publicly disclosed after a targeted attack took place. *Symantec* coordinated the disclosure of the attack and the vulnerability with *Microsoft* [16], and *Microsoft* released guidance to security vendors in the *Microsoft Active Protections Program* (*MAPP*) to help provide protection and also to monitor the threat environment until the update was fully tested and ready for release. The original attack only activated after a browser check for a vulnerable version. Following the disclosure, no attack activity was observed for about a week, although several proof-of-concept exploit code examples were published. Ten days into the disclosure, the attack was integrated into at least one exploit toolkit, launching attacks at any client (vulnerable or not) (Figure 5). The initial attacks were fairly limited in number and in geography, focusing mostly on South Korea, but spiked significantly just before the release of the update [17].

Although the disclosure of the targeted exploitation and vulnerability was coordinated, it's worth questioning whether it was premature. Although an attack had occurred in the wild, it was highly targeted. Given that an update was not available, did the disclosure of this vulnerability at that time do more harm than good? The publication of the exploit details (the PoCs) added fuel to the fire, leading to attacks in South Korea before the update came out. Publicly disclosing exploit details, rather than using private channels, provides a non-discriminate aid to good and bad actors. The data shows that consumers may suffer consequences.

In early February 2009, rumours started to emerge that a vulnerability in *Adobe Acrobat* was being exploited in targeted attacks. These rumours caused a great deal of concern among users. On 19 February, *Adobe* confirmed that a vulnerability existed (CVE-2009-0658 [18]) and that an update was under preparation [19]. Continued concern about the need to detect attacks targeting this vulnerability prompted a security researcher to post technical details to a public blog the very next day (Figure 6). Although the purpose of this disclosure was to help users protect themselves, it also had the effect of informing attackers on how to exploit the vulnerability. Widespread exploitation started immediately, although an update was not available for many days. How differently might this situation have unfolded if the fact that exploitation was occurring had been kept under wraps until an update was available?

A more recent example that followed a different path may help answer that question. CVE-2012-1856 is a vulnerability that was brought to *Microsoft*'s attention through a coordinated vulnerability disclosure [20]. At the time, targeted attacks were using the vulnerability. However,

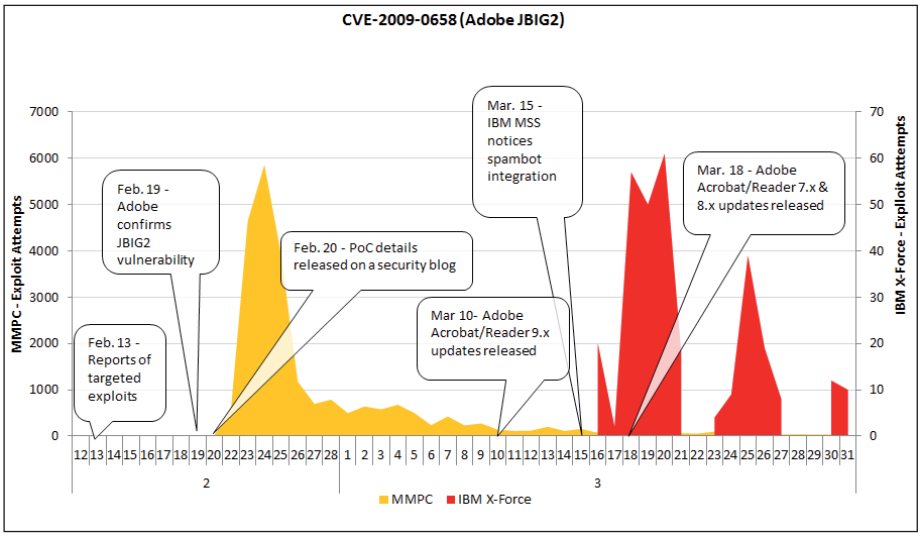

Figure 6: CVE-2009-0658.

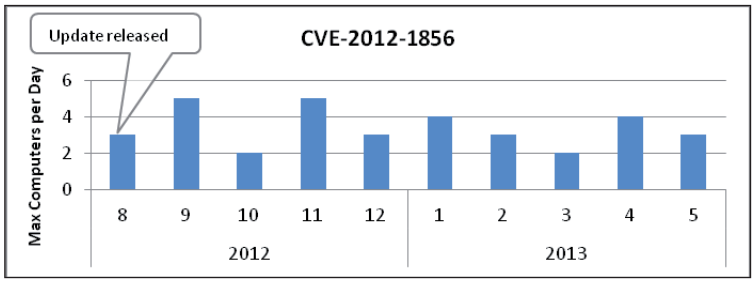

Figure 7: CVE-2012-1856.

Microsoft and the discloser held off on reporting the vulnerability until the update was available. Technical details about the vulnerability were provided through the *MAPP*, and some samples were shared in closed security groups. However, as of the time of writing, no details about the exploit have been publicly disclosed. Real-world exploitation of this vulnerability has not significantly escalated in spite of the fact that *Microsoft* gave this vulnerability a high exploitability index rating. To date, only traces of working exploits have been detected, with most detections related to researcher tests of proof-of-concept exploit documents (Figure 7).

A use-after-free vulnerability in *Microsoft Internet Explorer* provides another useful counter example. CVE-2011-0094 was first noticed in targeted attacks on 10 January 2011 [21]. Although information about the vulnerability was publicly posted on a somewhat obscure research website, it went unnoticed from the public eye for several months and, concurrently, little exploit activity occurred while *Microsoft* worked to prepare an update. In March, rumours surfaced in the security community about a new vulnerability in *Internet Explorer*, but exploitation outside of the targeted context was not observed until 8 April – four days before an update for the issue became available (Figure 8). Most of the attack activity was constrained to a particular geographic region. Because information about targeted exploitation of this vulnerability was not widely disseminated, events moved more slowly than in the *Adobe* JBIG2 example (CVE-2009-0658 [18]).

These examples illustrate that vulnerabilities may remain undisclosed to the public for long periods of time while targeted attack activity is occurring. The vulnerabilities uncovered in the *Symantec Research Labs* study were found to have been in the wild for an average of 312 days before disclosure [6]. The examples also illustrate that public disclosure of information about the vulnerabilities, including the fact that attacks are targeting them, can result in an increase in attack activity.

4. THE TIMING OF EXPLOITATION DISCLOSURE

Ultimately, the question of whether or not it makes sense to inform the public about active exploitation of a vulnerability rests on whether or not the positive consequences of informing the public outweigh the negative ones. The negative consequences relate to how widespread the exploitation activity is. If only a limited number of attackers are currently aware of a particular vulnerability, then public disclosure can result in a significant increase in attack activity. The positive consequences ultimately depend on how actionable the information is for users and administrators – the more that users can (and will) do with the information that is disclosed, the more benefit the disclosure has. By considering these two dimensions – the scope of present attack activity and the amount of information presently available about the vulnerability – we can determine whether public disclosure makes sense.

A. The scope of present attack activity

The scope of present attack activity can be categorized into three general states as follows:

- **Targeted:** attacks are focused on a specific organization or perhaps a small collection of organizations.
- **Limited:** relatively low numbers of attacks are occurring, could be predominantly affecting one region or industry.
- **Broad:** indiscriminate attacks are occurring across the Internet, targeting multiple geographic regions and industries.

Over time, exploitation activity associated with a particular vulnerability can increase. Vulnerabilities that start out only being used in highly targeted attacks can move to being used on a limited basis in a particular region, and eventually they might become broadly exploited throughout the Internet. Often, the public disclosure of information about the vulnerability is the catalyst that results in an increase in exploitation.

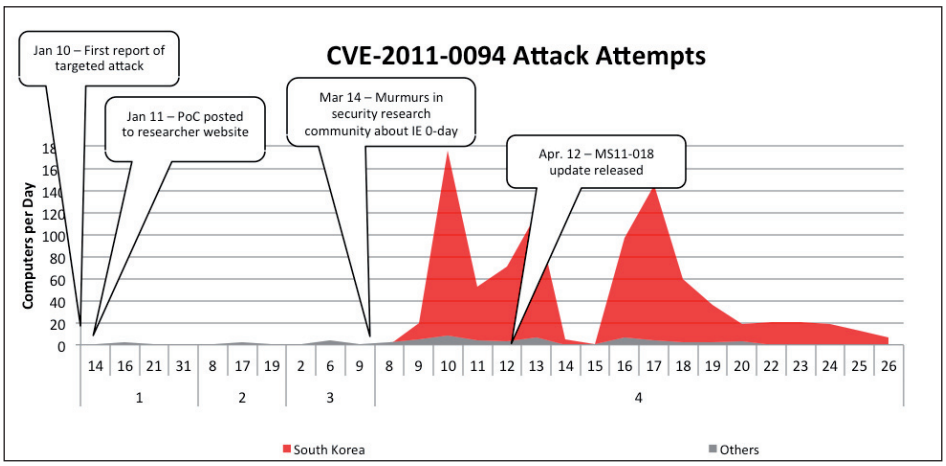

Figure 8: CVE-2011-0094 attack attempts.

It is important to recognize the difference in the nature of the threat actors involved in each stage. Threat actors that have the capability and motivation to discover and exploit new security vulnerabilities have some need for the level of stealth and access that comes with those sorts of attacks. Once a vulnerability is publicly disclosed, it may be less

valuable to such a threat actor. It is also worth noting that the sort of security organizations that are equipped to respond to sophisticated targeted attacks are more capable than the average end-user.

Consider, in light of these facts, the impact of announcing to the public that targeted attacks have been discovered for a security vulnerability that was previously undisclosed. This act has three consequences. First, the threat actor launching the initial targeted attacks may be less inclined to use that vulnerability now that it has been publicly disclosed and people are looking for attacks that target it. Second, sophisticated security teams equipped to deal with targeted attacks may be able to use this information to mitigate the vulnerability. Third, groups interested in launching widespread attacks on the Internet may become aware of the vulnerability and may start to target it. Without updates, there may be little that average end-users are able to do to protect themselves against these attacks. Even when updates become available, it may take some time for end-users to become aware of them and get them installed.

The third consequence also comes into play when vulnerability exploitation is prolific but constrained to a limited population, such as a particular geographic region. Threat actors on the Internet are not all working together, so disclosure, even in the case of limited exploitation, can still do harm by making attack activity more widespread.

An example is CVE-2012-4681, a vulnerability that was exploited in the wild for some time before an update was released [22]. Initially, exploitation was limited, until a security vendor noticed it in the wild and published details about the issue. Proof-of-concept exploit code was released within a week, exploitation soared from limited (around 50 computers per day) to widespread (15,000 computers per day) (Figure 9).

Typically, organizations that are equipped to respond to sophisticated, targeted attacks participate in private threat-indicator sharing relationships with peer organizations. These private back channels may be a mechanism by which information about new security vulnerabilities could be shared without tipping off attackers who are interested in launching widespread attacks. CVE-2012-1856 [20], discussed in the previous section, is a good example where details were shared in closed security groups, not publicized on the Internet, and not widely exploited. Using these closed

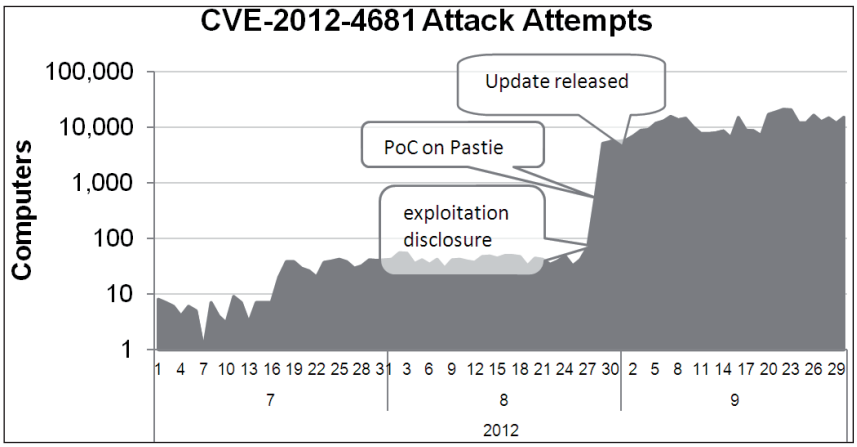

Figure 9: CVE-2012-4681 attack attempts.

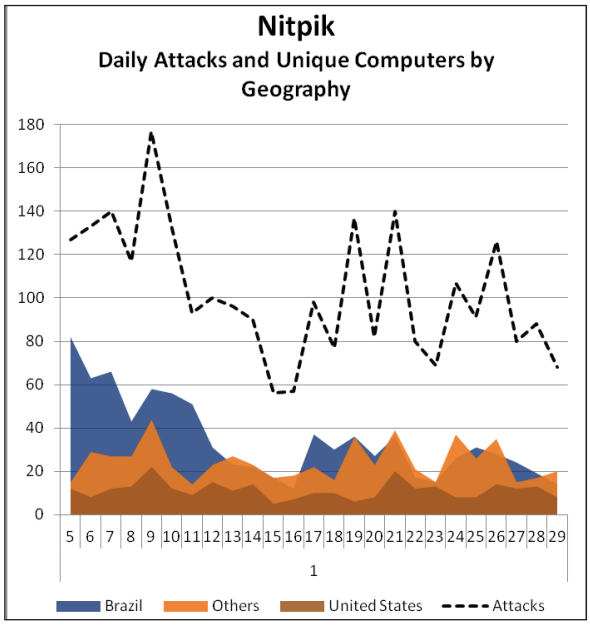

Figure 10: Nitpik attacks.

groups might have value in slowing the spread of exploitation even if one assumes that some may have been infiltrated by sophisticated threat actors who launch targeted attacks.

Furthermore, private coordination may be necessary in order to collect accurate information about the amount of attack activity occurring on the Internet at a particular time. Many organizations lack the capabilities required to determine how broadly a particular vulnerability is being exploited. Private networks can connect security professionals with software vendors and security companies who have the global visibility required to assess the threat situation on the Internet in real time. Those organizations, working together, can make a determination as to the urgency of releasing information about a particular vulnerability to the general public. This methodology might not reach all possible stakeholders that could benefit from knowledge about the vulnerability (and could unintentionally include bad actors), but it would balance the ability to monitor the threat situation and react accordingly without the consequences of full disclosure.

In assessing the scope of exploitation that is occurring in the wild, it is also important to avoid being misled by activity that can look like in-the-wild exploitation but is not. Proof-of-concept files that security researchers are testing or have distributed may show up in data regarding detections on the Internet. Malformed files may also accidentally trigger an issue in the wild without the knowledge of or intent to exploit the vulnerability. These kinds of events should not prompt analysts to conclude that the situation has escalated. Unexpected in-the-wild detections should be verified.

One example involved attacks associated with a small family of malware code-named Nitpik (Figure 10). Upon coordinating with the

vendor of the targeted software program, it was discovered that 'exploitation' activity which appeared to target a new vulnerability was actually a coding error – essentially an unintentional zero-day. Although the sample found in the wild crashed fully updated software, that piece of the code didn't deliver a payload.

If the information about this attack activity had been disclosed without this level of confirmation, it could have been used for true, successful malicious intent.

B. The disclosure state of the vulnerability

The public disclosure that exploitation is occurring may have different consequences depending on how much information is already available regarding the vulnerability. The more information that is available to the public about actions they can take to remediate the vulnerability, the more value that information about the exploitation has. The fact that a new vulnerability is being targeted could be disclosed before details of the specific vulnerability in question are disclosed, in conjunction with public disclosure of the vulnerability, or after the vulnerability has been disclosed. Let us consider each scenario in turn.

Disclosure of exploitation before vulnerability disclosure

Often, computer network breaches are disclosed without any information regarding the specific security vulnerability that was targeted. Legal regulations may require organizations to inform the public that data was compromised, but rarely do they ask organizations to explain how it happened. This is unfortunate because this knowledge of real-world security issues helps organizations to prioritize mitigation work. Without specific information about the nature of the vulnerabilities exploited in an attack, other organizations won't know where to focus their efforts in order to avoid a similar fate.

Sometimes the public is informed that a new vulnerability is being exploited but information about that vulnerability isn't disclosed. Unfortunately this can cause panic, particularly if mitigation advice isn't provided, because people know there is a threat out there but they don't know what to do about it. The real cost is incorrect actions taken due to the panic: shutting down services, admins working extra hours, deploying mitigations, etc. Consider the example of SockStress, a resource exhaustion denial-of-service tool that was announced before the techniques it used were publicly disclosed. News media accounts warned that the tool was very dangerous, and even included pictures of nuclear explosions and talk of a complete meltdown of the Internet. However, as no technical information was disclosed, there was no way for the security community to independently verify these claims. In such circumstances, even if mitigation advice is available, it might not be trusted unless enough technical information about the vulnerability is available to validate the need for the mitigation.

Given these factors, it may be best to wait until a vulnerability is disclosed before informing the public about exploitation activity that is targeting it.

Disclosure of exploitation in conjunction with vulnerability disclosure

Disclosure of the vulnerability in conjunction with disclosure of the fact that the vulnerability is being exploited allows the public to be fully informed about the nature of the threat. The key challenge is determining when it is safe to release this information. The right timing depends on the current level of exploitation as well as the state of the vulnerability remediation effort. Specific guidance will be provided in Section 5 of this paper. Generally speaking, if exploitation is targeted and no update is currently available, it may make sense to hold off on disclosure until the update is ready.

Regardless, coordination of this disclosure with the affected software vendor makes sense in every situation. It is important to get accurate information about mitigation into any disclosure of information about a vulnerability, and the affected software vendor will have more precise information about mitigation techniques and more direct access to customers who may be impacted than any other organization.

Disclosure of exploitation after vulnerability disclosure

It makes sense to publicly disclose information about exploitation, even in the case where the vulnerability has already been disclosed. As we have illustrated, awareness of attack activity can help users of a particular software application prioritize their mitigation steps. This information can also help security product vendors validate that their protections are working correctly.

However, even in these cases it is important to coordinate disclosure with the affected software vendor before going public. The *Microsoft Malware Protection Center* encountered a good example of how that kind of coordination can be beneficial when it released a blog post [23] on a significant increase in Java exploitation (Figure 11).

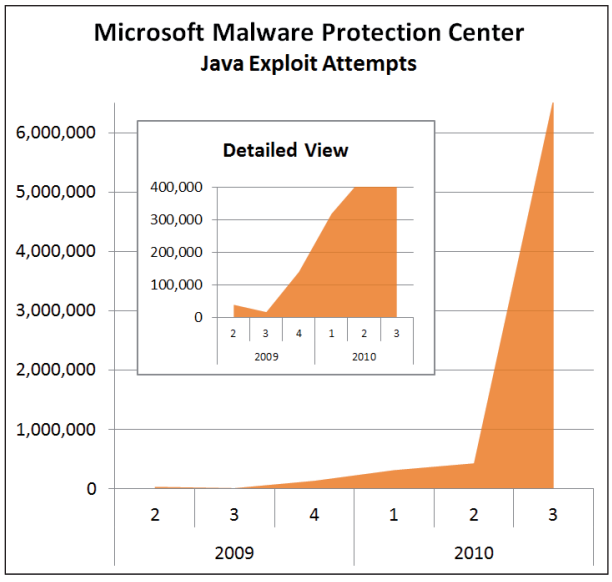

Figure 11: Java exploit attempts.

The *Microsoft Malware Protection Center* provided *Oracle* with a preview of the post. The draft post failed to mention that all of the Java vulnerabilities being exploited had an update available – a fact known to security researchers but not necessarily known to the everyday reader. This one detail was the critical piece of information needed to put the power into the hands of the readers – 'yes there is cause for alarm

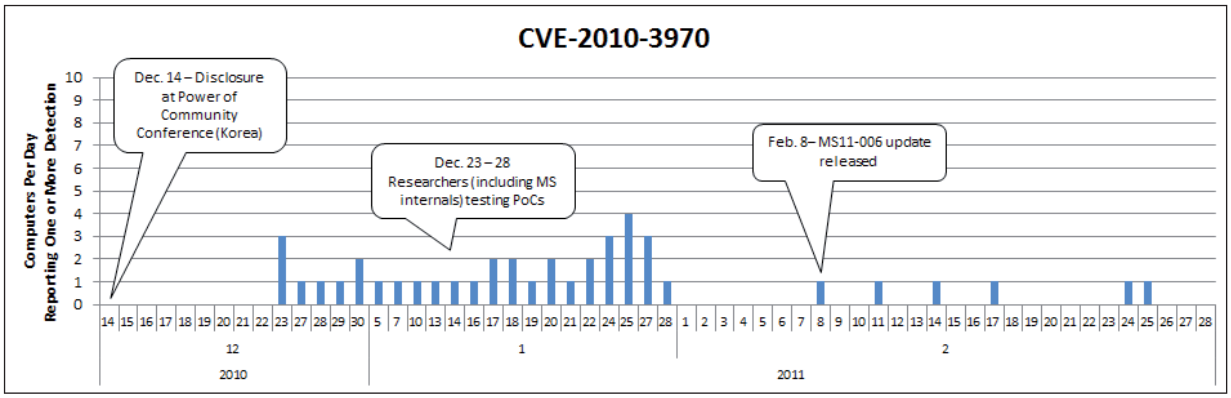

Figure 12: CVE-2010-3970.

because of the spike in attack activity, but apply the updates and all will be fine.' *Oracle* caught this very important omission before *Microsoft* published its blog post, it was corrected, and readers benefited from it.

It is also important to recognize that even in situations where a vulnerability has been publicly disclosed, there might not yet be widespread awareness within attacker communities regarding the opportunity that the vulnerability represents. CVE-2010-3970 [24] was a remote code execution vulnerability in *Microsoft Windows Picture and Fax Viewer Library* that was publicly disclosed at a security conference in December 2010 [25]. Although this vulnerability had been publicly disclosed nearly two months before an update became available, there was no real in-the-wild attack activity until the day that *Microsoft* released an update that addressed it (Figure 12). Had broader communication occurred before the update was available, it might have tipped off attackers.

5. PUTTING IT TOGETHER – GUIDANCE FOR EXPLOITATION DISCLOSURE

General Guidelines for Exploitation Disclosure

	Unknown, 0-day/No Update	Known, No Update or Workaround	Known, Workaround available	Known, Update available
Targeted	Coordinate and wait for updates	Coordinate and wait for updates	Coordinate and wait for updates	Coordinate, but don't wait
Limited	Coordinate and confirm it	Coordinate, maybe wait	Coordinate, maybe wait	Coordinate, but don't wait
Broad	Coordinate, but don't wait	Coordinate, but don't wait	Coordinate, but don't wait	Coordinate, but don't wait

Figure 13: Guidelines for exploitation disclosure.

Figure 13 summarizes our general advice regarding whether or not to publicly disclose the fact that a vulnerability is being exploited in the wild, given the amount of exploitation occurring at a particular time and the disclosure state of the vulnerability at that time. Some of these cases are straightforward. Others require a careful look at the specific circumstances involved. (And, with any case, situational judgement may override the general recommendation.)

If a vulnerability is being exploited broadly on the Internet, it almost always makes sense to disclose that fact to the public so that administrators can take defensive actions, no matter what the disclosure state of the vulnerability is at that time.

However, this sort of public disclosure should always be coordinated with the affected software vendor in order to make sure that mitigation advice is as accurate as possible and to make sure that the information reaches all of the right people.

If a vulnerability is only being utilized in targeted attacks, on the other hand, in general it makes sense to hold off public disclosure of that fact until an update is available from the software vendor, because publicly disclosing the fact that exploitation is occurring may attract attackers to target the vulnerability before the update is ready.

Limited exploitation provides a more challenging set of circumstances, particularly given that when a vulnerability is first discovered in use in the wild, it may be hard to determine exactly how widespread the attack activity is. The first step that someone who discovers such a vulnerability should take is to work with the affected vendor to confirm the amount of attack activity that is occurring in the wild.

Similar to the case of targeted exploitation, disclosure of the fact that attacks are occurring can accelerate attack activity. A judgement call needs to be made based on how soon updates will be available and how quickly the attack activity is increasing. In considering this situation, it is also important to keep in mind that end-users are more likely to deploy an update than a workaround. Even if a workaround is available, the number of people who stand to be harmed by an increase in attack activity will often be larger than the number of people who stand to benefit from a disclosure.

Given those circumstances, in most cases it makes sense to wait for an update to become available before disclosing any additional information either about the vulnerability or about the attacks that are targeting it, unless the update is going to take a long time to release or the amount of attack activity is increasing rapidly.

Consider the example of CVE-2010-2568 [26], the .lnk vulnerability that was exploited by the Stuxnet worm. This vulnerability was first discovered as part of Stuxnet, a worm used in targeted attacks that would seem to meet the criteria for immediate coordinated disclosure. Under normal circumstances, an Internet worm would constitute broad exploitation, and under our framework this would suggest that immediate, coordinated disclosure is the right thing to do. However, as you can see from Figure 14, the disclosure of this vulnerability before an update was available resulted in immediate increases in attack activity, as other worm

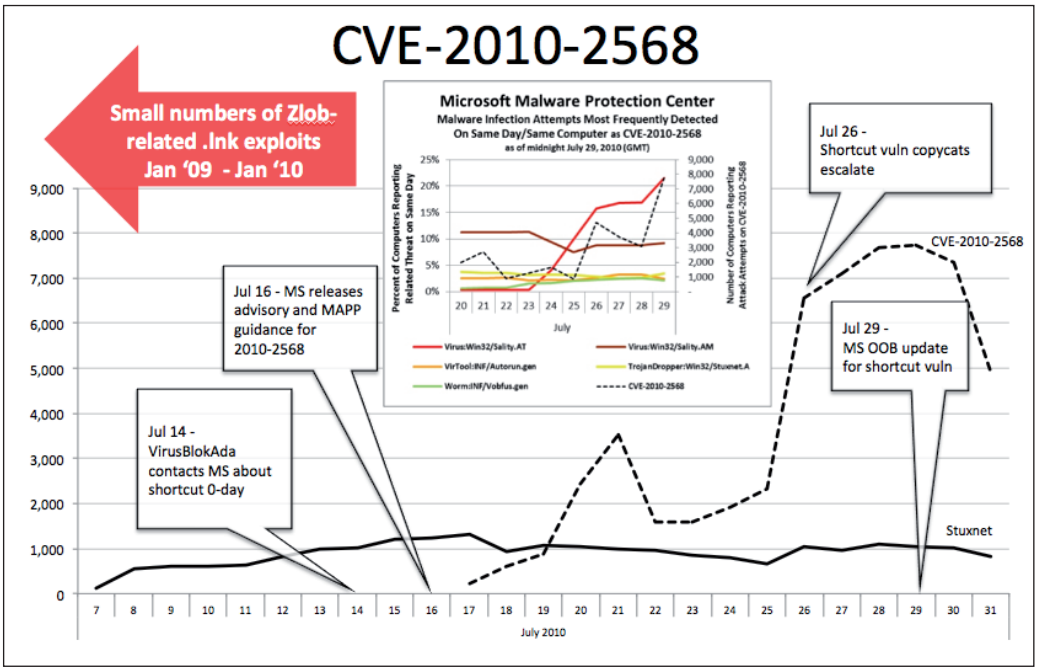

Figure 14: CVE-2010-2568.

operators moved to incorporate the attack into their malware [27].

Although Stuxnet was self-replicating malware, the rate at which it was spreading was slow relative to typical Internet worms. Furthermore, the update that addressed this vulnerability was made available very rapidly – two weeks after disclosure. Given the limited spread of the malware in this case and the rapid delivery of the update, in hindsight, holding back on this disclosure for a couple of weeks might have been a better call. However, this is a close-case presented for the purpose of illustrating how difficult these decisions can be in practice.

Whenever you publicly disclose information about vulnerabilities or in-the-wild exploitation, it makes sense to keep the following advice in mind:

- Put hashes (MD5, SHA1, etc.) of the malware samples you've seen in blog posts to help vendors with identifying samples and sample detection.
- Avoid providing exploit details that might help copycat attackers.
- Include the CVE number for the vulnerability or go back and add it later if it is not assigned at the time that you publish. This makes your disclosure easy to search for and associate with the vulnerability.
- Reference the specific product updates or workaround information for the vulnerabilities in question – people need to know this information.
- Consider providing data to industry partners, either independently or through a program like *MAPP*, before the data is disclosed to the general public.

6. CONCLUSIONS

We have explained the difference between vulnerability disclosure and exploitation disclosure. We have established that public disclosure of the fact that a vulnerability is being exploited is valuable because knowledge of attack activity can help end-users prioritize mitigation activity, it can help software vendors prioritize the development of updates, and it can help security vendors validate their protections.

We have also established that exploitation disclosure can do harm, because public knowledge that a vulnerability is exploitable attracts attack activity. We have presented a framework that can guide the timing of exploitation disclosure by considering the amount of attack activity occurring at a particular time as well as the state of the vulnerability remediation effort.

Our most important conclusion is that public disclosure of the fact that a vulnerability is being exploited should be coordinated with the affected software vendor. Only the affected vendor can ensure that the correct mitigation information is included in a disclosure and that the users who are impacted are properly informed.

REFERENCES

[1] Microsoft. Microsoft Security Intelligence Report Volume 14. April 2013. http://download.microsoft.com/download/E/0/F/E0F59BE7-E553-4888-9220-1C79CBD14B4F/Microsoft_Security_Intelligence_Report_Volume_14_English.pdf.

[2] Schneier, B. Full Disclosure of Security Vulnerabilities a 'Damned Good Idea'. January 2007. https://www.schneier.com/essay-146.html.

[3] Culp, S. It's Time to End Information Anarchy. October 2001. http://www.angelfire.com/ky/microsfot/timeToEnd.html.

[4] Christey, S.; Wysopal, C. Responsible Vulnerability Disclosure Process. February 2012. http://www.cert-ist.com/documents/Document_Cert-IST_000074.txt.

[5] Microsoft. Coordinated Vulnerability Disclosure. https://www.microsoft.com/security/msrc/report/disclosure.aspx.

[6] Bilge, L.; Dumitras, T. Before We Knew It: An Empirical Study of Zero-Day Attacks In The Real World. In ACM Conference on Computer and Communications Security (CCS), Raleigh, NC, October 2012.

[7] Cross, T. When can alerting the public about exploitation do more harm than good? IBM FrequencyX Blog. 15 July 2011. http://blogs.iss.net/archive/publicexploitation.html.

[8] CVE-2010-1885. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-1885.

[9] Stewart, H. Attacks on the Windows Help and Support Center Vulnerability (CVE-2010-1885). Microsoft Malware Protection Center. 30 June 2010. http://blogs.technet.com/b/mmpc/archive/2010/06/30/attacks-on-the-windows-help-and-support-center-vulnerability-cve-2010-1885.aspx.

[10] Stewart, H. Update on the Windows Help and Support Center Vulnerability (CVE-2010-1885). Microsoft Malware Protection Center. 13 July 2010. http://blogs.technet.com/b/mmpc/archive/2010/07/13/update-on-the-windows-help-and-support-center-vulnerability-cve-2010-1885.aspx?Redirected=true.

[11] CVE-2011-0611. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-0611.

[12] Adobe. Security Advisory for Adobe Flash Player, Adobe Reader and Acrobat. 11 April 2011. http://www.adobe.com/support/security/advisories/apsa11-02.html.

[13] Parkour, M. Apr. 8 CVE-2011-0611 Flash Player Zero day – SWF in DOC/ XLS – Disentangling Industrial Policy... Contagio. 11 April 2011. http://contagiodump.blogspot.com/2011/04/apr-8-cve-2011-0611-flash-player-zero.html.

[14] CVE-2010-3333. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-3333.

[15] Finones, R. Targeted Attacks Against Recently Addressed Microsoft Office Vulnerability (CVE-2010-3333/MS10-087). 29 December 2012. http://blogs.technet.com/b/mmpc/archive/2010/12/29/targeted-attacks-against-recently-addressed-microsoft-office-vulnerability-cve-2010-3333-ms10-087.aspx.

[16] Microsoft. Microsoft Security Advisory (2458511). 3 November 2010. http://www.microsoft.com/technet/security/advisory/2458511.mspx.

[17] Stewart, H. CVE-2010-3962 – The weekend warrior. Microsoft Malware Protection Center. 9 December 2010. http://blogs.technet.com/b/mmpc/archive/2010/12/09/cve-2010-3962-the-weekend-warrior.aspx.

[18] CVE-2009-0658. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-0658.

[19] Adobe. Security Updates available for Adobe Reader and Acrobat versions 9 and earlier. 19 February 2009. http://www.adobe.com/support/security/advisories/apsa09-01.html.

[20] CVE-2012-1856. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-1856.

[21] CVE-2011-0094. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-0094.

[22] CVE-2012-4681. http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-4681.

[23] Stewart, H. Have you checked the Java? Microsoft Malware Protection Center. 18 October 2010. http://blogs.technet.com/b/mmpc/archive/2010/10/18/have-you-checked-the-java.aspx.

[24] CVE-2010-3970. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-3970.

[25] Moti; Hao, X. A Vulnerability in My Heart. Power of Community, Korea, 2010. http://powerofcommunity.net/poc2010/moxu.pdf.

[26] CVE-2010-2568. http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2010-2568.

[27] Stewart, H. Stuxnet, malicious .LNKs, ...and then there was Sality. Microsoft Malware Protection Center. 30 July 2010. http://blogs.technet.com/b/mmpc/archive/2010/07/30/stuxnet-malicious-lnks-and-then-there-was-sality.aspx.