info@thinkst.com
research@thinkst.com
http://www.thinkst.com

Client:	**Marco Slaviero**
Date:	March 2011
Ad-hoc Information Update -TAH02	
PWN2OWN - What it Means to You	

Table of Contents

PWN2OWN & You

Introduction

March 2011 saw the 12th CanSecWest conference in Vancouver. CanSec is a quality conference, but these days generates as many headlines for its PWN2OWN contest, as for its talks. The contest seems to be quite polarising, with some security luminaries calling the contest "*harmful*" [1] and others singing its praises. Does it have value for you? Should you care about the results?
This is a quick ThinkstScapes update with some of the salient points and interesting take-aways.

This is an Ad-Hoc update created and distributed under the ThinkstScapes subscription service. Ad-Hoc updates are sent out to customers throughout the year as events worthy of notice transpire. Ad-Hoc updates are usually brief, bursty and bustled out while events unfold.

Overview

PWN2OWN has been held as a side-show to CanSecWest since 2007, when Dragos Ruiu setup 2 MacBook Pro's, and aimed to "*see how well a default OSX install really does in a room full of security researchers*".[2] Although the contest rules have had slight modifications in the past 5 years, the essence remains the same:

> The first person to compromise one of the designated (fully patched) target machines, gets to keep it (along with a cash prize from TippingPoint).

In 2009, mobile phones were added as target machines and from 2010 these devices too were compromised.

Figure-1 - 4 time PWN2OWN Winner Charlie Miller

Critics of PWN2OWN assert that it is designed to generate headlines, and that we learn very little from the contest. Jeff Jones (Microsoft) is reported to have said that the contest is "*simplifying security to the point of uselessness*"[3].

We believe that there are indeed lessons for us in PWN2OWN, but that these might not be immediately obvious.

---

[1] http://lcamtuf.blogspot.com/2011/03/pwn2own-considered-somewhat-harmful.html

[2] http://archives.neohapsis.com/archives/dailydave/2007-q1/0290.html

[3] http://news.softpedia.com/news/Charlie-Miller-Wins-Pwn2Own-Again-Thanks-to-Safari-Flaw-107269.shtml

Quick Summary of Previous Contests

In 2007, famed security researcher Dino Dai Zovi put together an exploit which abused Safari (Apples default browser) to win the contest. This was well publicised, and was the last time that a vulnerability would be hunted, discovered and exploited strictly during the contest. Future contest winners would have months to prepare, and some would even make use of bugs stored for over a year in private collections.

Year	Target	Compromised By
2007	MacBook (OS X, Safari)	Dino Dai Zovi / Shane Macaulay
2008	Fujitsu (Vista SP1)	Shane Macaulay
	Macbook Air (OSX 10.5.2)	Charlie Miller
	Sony Vaio (Ubuntu 7.10)	-
2009	Internet Explorer 8 (Windows 7)	Nils
	Safari (OS X)	Charlie Miller & Nils
	Firefox (OS X)	Nils
	Firefox (Windows 7)	-
	BlackBerry, Android, iPhone, Symbian, Windows Mobile	-
2010	Safari (OS X)	Charlie Miller
	Internet Explorer 8 (Windows 7)	Peter Vreugdenhil
	Firefox (Windows 7)	Nils
	iPhone 3GS	Vincenzo Iozzo & Ralf Philipp Weinmann
	Google Chrome (Windows 7)	-
	BlackBerry, Android, Nokia	-
2011	Safari (OS X)	Vupen (team)
	Internet Explorer 8 (Windows 7)	Stephen Fewer
	iPhone 4	Charlie Miller & Dion Blazakis
	BlackBerry	Vincenzo Iozzo, Willem Pinckaers, Ralf Philipp Weinmann
	Firefox (Windows 7)	-
	Google Chrome (Windows 7)	-
	Android, Windows Phone7	-

Time to Exploit

An SC Magazine headline reads: "*Safari and IE8 broken in moments at pwn2own*"[4]. (PC World reported that "*Safari fell in the blink of an eye.*"[5])

While it is technically true that the exploit took only a few moments to compromise the machine, it's obvious to anyone in the industry that the time an exploit runs for, has no correlation to finding a vulnerability or building a reliable exploit for it. *(Anecdotally, it seems like 6 weeks was about the average time spent per target).*

> ***Quick take-away: Time taken to exploit the target on the day, has nothing to do with how long it took to discover the vulnerability (or to write the exploit)***

Mobile Targets

It is well worth noting that the founder of CanSec started PWN2OWN to prove empirically that a platform was vulnerable[6] (despite the fact that it was seldom exploited in the wild). Fast forward 4 years, and people have similar security assumptions around mobile devices (due largely to their locked down nature). When mobile devices were added in 2009, all devices escaped untouched.

The very next year however, the iPhone was taken in fine style, and this year both the iPhone and the BlackBerry fell due to browser bugs.

> ***Quick take-away: The closer mobile phones become to full blown machines, the more attractive they become to attackers.***

The Browser as the Weakest Link

Client-side exploits were long considered the red-haired step child of vulnerability research, and this was reflected in the original contest rules. When PWN2OWN first launched, attacks that required a user to click on a link were considered a second-class victory. It is really interesting then, that by the end of 2008, the contest morphed completely, into a browser hacking game.

This is an interesting turn that few people pay attention to.

The browser is a complex piece of software which is forced to deal with parsing multiple file formats while hosting a number of scripting languages. In our 2010 BlackHat USA presentation on memory corruption vulnerabilities[7], we pointed out that the fine grained control that an attacker has over a clients browser is the perfect vehicle to manipulate the targets address space, allowing him to bypass memory trespass mitigations. The increasing demand for web applications that mimic native applications means that this trend is not going away, and browser complexity keeps growing.

> ***Quick take-away: Browsers are a super attractive target for memory trespass attacks providing a number of input vectors and fine grained control of the targets address space.***

---

[4] http://www.securecomputing.net.au/News/250886,safari-and-ie8-broken-in-moments-at-pwn2own.aspx

[5] http://www.pcworld.com/businesscenter/article/221848/what_pwn2own_tells_us_about_browser_security.html

[6] http://archives.neohapsis.com/archives/dailydave/2007-q1/0290.html

[7] http://blog.thinkst.com/2010/08/blackhat-2010-slides-paper-rest.html

Your Current Exposure

CanSec attracts excellent researchers, but we have to keep in mind that the attending researchers are a small percentage of the total pool of vulnerability researchers. Even the attending researchers sometimes lack the incentive to participate, calling the prize money insufficient: "[IE flaws] are worth way more money—more people use them,"[8].

> ***Quick take-away: The real threats, are not the guys attending the conference and disclosing their bugs.***

With PWN2OWN, a non-technical user/observer goes through the following stages:

Time

[1] User feels relatively secure browsing the Internet with his fully patched browser.

[2] PWN2OWN is held, and the fully patched browser is exploited (in the public eye).

[3] Vendor starts working on a patch.

[4] User feels very exposed knowing that his browser is currently not perfectly safe.

[5] Vendor issues patch for PWN2OWN vulnerability.

[6] User feels relatively secure browsing the Internet with his fully patched browser.

Fearing the security of our browser, because the attack has been demonstrated at CanSec, implies that we previously assumed our browser was not vulnerable. The fact that the browsers are exploited consistently every year, makes this assumption obviously incorrect.

The contest simply shows that with about 2 months of dedicated work (or about $20k), it is possible to obtain an exploit for the (fully patched) browser you are currently using.

In some cases, the time to exploit is far less: (Consider Charlie Miller, who having won the contest with a Safari bug in 2008, saved a 2nd bug to use the following year.)

> ***Quick take-away: 0-day happens. We need to make sure we are planning for it in our network architecture and design***

Are you One 0day Away from the Worst Day of Your Work Life?

Attackers have long ago stopped aiming at public facing servers, and instead started concentrating on client side attacks. These attacks get the attacker directly onto the users machine, with the same level of access that the user has.

When we realised that our external servers were compromisable, we created DMZ's. We were effectively quarantining the hosts that were possibly tainted from the rest of the crown jewels. In the current landscape, the browser becomes the tainted, compromisable piece of the infrastructure, but very few of us have built in, a network tolerance for compromised browsers.

PWN2OWN shows us that a few thousand Dollars could secure a reliable browser 0day. The question you then have to ask is: "How many 0-days away is your company from the worst (work) day of your life?"

> ***Quick take-away: If our CEO/CFO/DBA is able to connect to the company jewels with the same machine he has just used to surf the open Internet, then we need to realise that our company jewels are just one exploit away from the opposition.***

---

[8] http://securitywatch.eweek.com/apple/mac_hacked_via_safari_browser_in_pwn2own_contest.html

Conclusion

Pwn2Own does indeed attract atrocious headlines, but also serves to confirm a pretty simple truth: **0day happens**.
For some reason, most organisations today ignore the fact that our Browsers are simultaneously one of the most exploitable and one of the most exposed, pieces of software on our machines.
If we do have crown jewels to protect, we need to make sure that these jewels, are not one 0-day away from being shared on the Internet.

This ad-hoc update was authored by {haroon;jameel}@thinkst.com. Please mail us if you have any questions, comments or thoughts in this regard.