

Memory Corruption, Exploitation, and You

Dino A. Dai Zovi < ddz@theta44.org>

Introduction

- # Me:
 - * Info Sec Consultant / Researcher / Philosopher
- * This (experimental) presentation:
 - * Rants and raves in no particular order
- * You:
 - * Draw your own conclusions

Talking Points

- * "How many APTs Chinese Teenage Hackers (CTH) does it take to break into your network?"
- * "Lies, damn lies, and fuzzing statistics
- * "Vulnerabilities don't 0wn people, exploits 0wn people"
- * "Buffer overflows are so 1999"



IS THIS THE ADVANCED PERSISTENT THREAT?

OR, "HOW MANY CHINESE TEENAGERS DOES IT TAKE TO COMPROMISE YOUR NETWORK?"

Cyberwarriors or Cyberpunks?

- * Stop flattering yourself, your network is trivial to 0wn
 - * Your employees' names and e-mail addresses are enumerable on social networking sites?
 - * Your employees answer external e-mail and access Internet web sites on the same machine that they create or handle proprietary information?
 - * Are their e-mail addresses firstname.lastname@company.com?

ODay Attacks != H1N1

- Stop treating 0day attacks like H1N1
 - * People are getting sick with an unknown virus, we must respond to this incident
 - * Take anti-viral medication to treat infections
 - * We have developed an immunization shot for H1N1, everyone please go apply it to yourselves
- * Any hand-written malware will evade anti-virus
- * We don't have a cyber immune system yet

Public Health vs. Crime

- * Mass malware and botnets are an Internet public health problem (cyberhealth?)
 - * Opportunistic, low-skill and attention
- * Targeted attacks are a cybercrime problem
 - * Deterrence requires enforcement and prosecution (good luck on that!)
 - * In absence of those, prevention is best recourse

Prevention is Hard

- * Because the security industry isn't making the right products or tools
 - * No one bought the effective ones because they didn't understand them or couldn't justify them
- * Vulnerability and exploitability analysis is confusing
 - * What mitigations are enabled in this application?
 - * Are they effective? Have they been disabled?

Eat the Rich AV Vendors

- * Overheard outside RSA exhibition hall:
 - * "Vendor spent \$500k on their booth exhibit and it costs them \$90k to transport and set it up anywhere"
 - * They have too much money for not solving today's real-world problems
- * Why pay protection money to the mafia when you are still getting robbed every day?

Exploits Should Be Hard

- * And they are getting harder, but not hard enough
- * Mass malware increasingly turning to social engineering tactics instead (i.e. rogue AV)
 - * Misanthropingly effective
 - * Real anti-virus can handle this problem
- * Defending against advanced attackers requires advanced defenses

But, But, ASLR, DEP!

- * ASLR and DEP do a great job of making exploitation of serverside vulnerabilities impossible in the vast majority of cases
- * Low-integrity prevents writing, but not reading your sensitive docs and information
- * Scriptable client applications offer a much larger element of attacker control ("control surface")
 - * Yields more possibilities for evading ASLR
- * Code-reuse exploit techniques can be used to bypass DEP

Code-Reuse Exploits

- * Return-To-Libc (Solar Designer, 1997)
 - * Return into functions in Libc
- * Borrowed Code Chunks (Krahmer, 2005)
 - * Link returns to single-instructions
- * Return-oriented Programming (Shacham, 2007)
 - * Turing complete w/ compiler for C-like language

Tactics vs. Strategy

- * Malicious injected code is not the true problem
 - * It is only the most common exploitation tactic
 - * Code-reuse exploitation techniques don't need to inject any code, will reuse what is there
- * The strategy is to make the target application do unexpected things in a way useful to the attacker
 - * Unexpected/undesirable behavior is problem

My Sandbox Soapbox

- * Why does my browser need to be able to write to anywhere except for ~/Downloads?
- * Why do doc readers, IM clients, need to write files at all?
- * Multi-user DAC security model is ill-suited to the desktop
- * We need a new multi-application desktop security model
 - * Phones (iPhone and Android) already have this
 - * iPhone prevents injected code and app misbehavior



LIES, DAMN LIES, AND FUZZING STATISTICS OR, AN ARMY OF MONKEYS WITH GUNS IS PRETTY SCARY

Charlie Miller's Fuzzing Stats

- * "Dumb fuzzing"
- * 12-25% of unique crashes deemed "exploitable"
- * 33-50% of unique crashes deemed "exploitable" or "probably exploitable"



Miller, Charlie. "Babysitting an Army of Monkeys: An Analysis of Fuzzing 4 Products with 5 Lines of Python". CanSecWest 2010. http://securityevaluators.com/files/slides/cmiller_CSW_2010.ppt

Meditate On These Numbers

- * 300 file format parsers
- * 1,000,000 fuzz iterations
- * 1600 unique bugs
- * 200-800 likely exploitable vulnerabilities



Senior security program manager Brad Albrecht pointed out that Office 2010 supports around 300 file formats, and that fuzz testing allowed the company to find and fix some 1600 bugs during the development cycle. "Our tools are getting better," he said.

Withers, Stephen. "Fuzzing Detected 1600 Office 2010 bugs During Development", ITWire, July 15, 2010. http://www.itwire.com/business-it-news/security/40430-fuzzing-detected-1600-office-2010-bugs-during-development

Mmmm... Stale Bugs

- * Not all of those parsers (and their bugs) were freshly written for Office 2010
 - * Most likely, the vast minority are "fresh" bugs
- * Every past release of Office has at least those 200-800 latent exploitable vulnerabilities
- * They all aren't going to be discovered, reported, and patched anytime soon...

There is no client-side DMZ

- * Receive packet/request => code execution
 - * Place front-end servers on the DMZ
 - * Firewalls between front-end web servers and back-end app or database servers
- * Parse/display data => code execution
 - * Handle incoming data on internal corp network?

Workstation Attack Surface



ZDI-CAN-855 Microsoft	CVSS: 10	2010-06-30 (121 days ago)	2011-02-04
Discovered by: Anonymous			
ZDI-CAN-852 Mozilla Firefox	CVSS: 10	2010-06-30 (121 days ago)	2011-02-04
Discovered by: wushi of teamS09			
ZDI-CAN-843 RealNetworks	CVSS: 10	2010-06-30 (121 days ago)	2011-02-04
Discovered by: Anonymous			
ZDI-CAN-842 Adobe	CVSS: 10	2010-06-30 (121 days ago)	2011-02-04
Discovered by: Abdullah Ada			
ZDI-CAN-829 Microsoft	CVSS: 10	2010-06-30 (121 days ago)	2011-02-04
Discovered by: Anonymous			
ZDI-CAN-828 Apple	CVSS: 10	2010-06-30 (121 days ago)	2011-02-04
Discovered by: Anonymous			
ZDI-CAN-798 Microsoft	CVSS: 10	2010-06-30 (121 days ago)	2011-02-04
Discovered by: Anonymous			

No End in Sight

- * There is no end in sight to the supply of vulnerabilities in dominant client-side applications
 - * Waiting for or expecting them to be discovered, reported, and patched at the current rate is naive
- * You must learn to defend against the known-unknowns (Oday attacks in the wild)
 - * Assume that data parsed, interpreted, and/or rendered from the Internet will yield code exec



GETTING READY FOR PWN20WN 2011

OR, WHY ODAY ATTACKS MAKE RESPONSIBLE DISCLOSURE IRRELEVANT

Vulnerability Handling

- * We need to debate "vulnerability response" not "responsible disclosure"
 - * "Responsible disclosure" presupposes many decisions, judgements, and interests
- * Was created for 2002's Internet, but now it's 2010
 - * Many of ZDI's "Upcoming Advisories" could enable an "Aurora"-style attack if exploited

Are We Chasing Our Tail?

- * Is the vulnerability disclosure status quo:
 - * awesome?
 - * sufficient?
 - * irrelevant?
 - * a distraction at best?
 - * enabling an addict?

Vulnerabilities vs. Exploits

- * A vulnerability never 0wned anyone, an exploit did
 - * There are more people that can find vulnerabilities than can write reliable exploits
 - * Count number of ZDI vulnerability contributors vs. PWN2OWN contestants past and present
 - * A minority of vulnerabilities have the potential to be turned into a dangerous exploit

Exploits Matter

- * OSVDB query for remote vulnerabilities in 2009
 - * ~1000 potential code/command execution
- * Manual analysis of exploit kits, incidents, etc.
 - * 40 exploits observed being used in the wild
 - * Most copied from Milworm with few changes
 - * Comment out skape/skywing DEP bypass

Bugs for Bosch

- * "Google attack highlights 'zero-day' black market" (AP, 1/29/2010)
 - * "I basically had to make a choice between doing something that would protect everybody and remodeling my kitchen — as terrible as that is, I made that choice, and it's hard," Miller said. "It's a lot of money for someone to turn down."
- * Adobe JBIG2 exploit was sold for \$75k (Twitter, I think)
 - * Reporting bug responsibly feels like a million bucks!

\$75K is a Lot of Food

- * \$75K = ~ \$512K CNY
- * Average yearly salary for a Software Engineer in China is \$90K CNY
 - * http://www.payscale.com/research/CN/ Country=China/Salary
- * Would you "do the right thing" for free when you could "do the wrong thing" for 5-6 years salary?

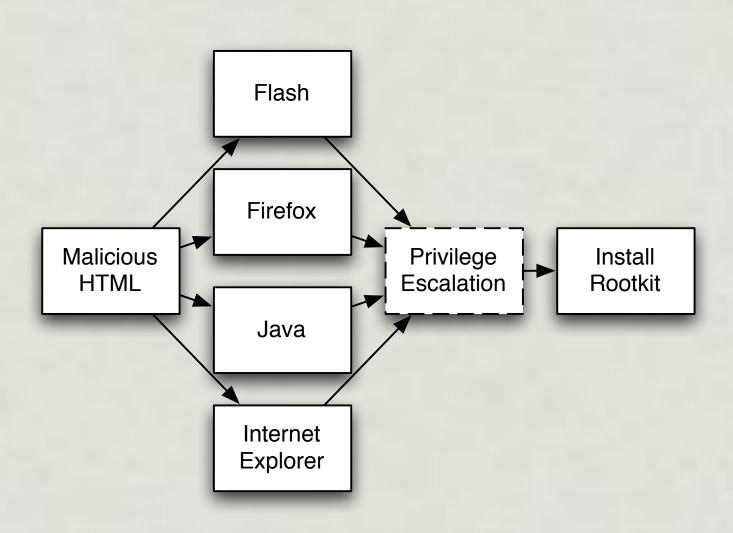
Fighting ODay Exploits

- * Make them illegal!
 - * Right... best of luck with that (WTO sanctions?)
- * Make a transparent, open, legitimate market!
 - * Vendors will never pay or play along
- * Make them ineffective!
 - * Now you're onto something...

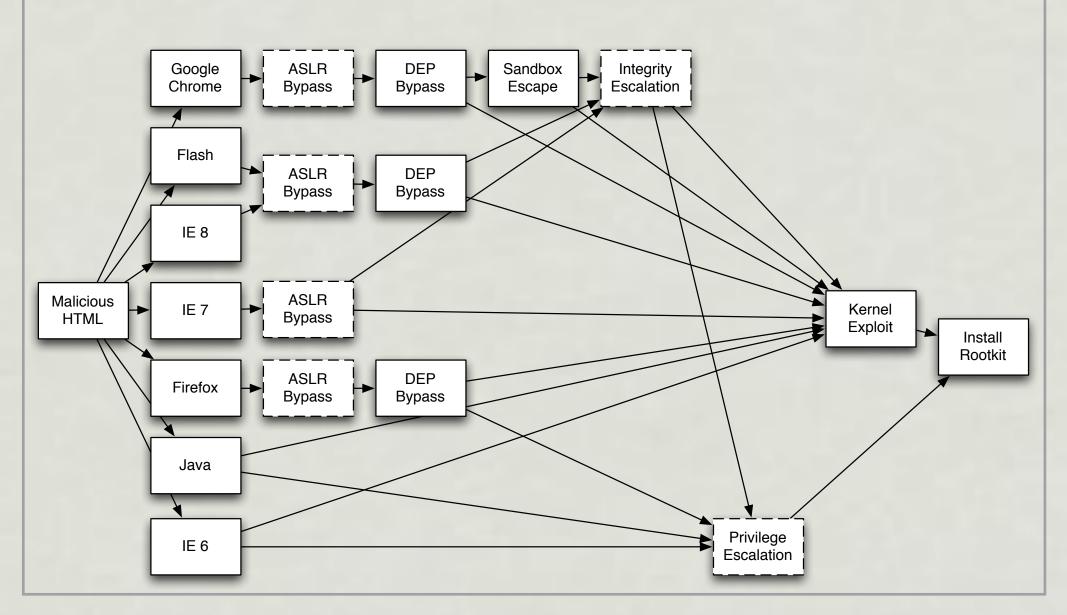
One ODay Ruins Your Day

- * One Oday browser or document reader exploit is a skeleton key for everyone's side doors
 - * The front door has layered firewalls, DMZs, hardened servers, ingress/egress filtering
 - * Client desktops are a wilderness of unmanaged or barely managed systems with software handling untrusted data as Administrator
 - * Client desktops have unlimited internal access

Browser Attack Graph c. 2005



Modern Browser Attack Graph





STOP CALLING THEM BUFFER OVERFLOWS!

UNLESS A BUFFER IS ACTUALLY BEING OVERFLOWN (INCREASINGLY RARE)

Vulnerability Terminology

- * Buffer overflow
 - * What about out-of-bounds array indexes?
- * Arbitrary code execution
 - * What about Solaris telnetd bug => auth bypass
- * Memory corruption
 - * What about use-after-free?
 - * What about memory disclosure vulnerabilities?

Type Safety

- * All of these vulnerabilities are failures of type safety
- * C/C++ are not memory-safe or type-safe
- * Type-safe languages only have these problems when their implementations, written in unsafe languages, have these vulnerabilities
 - * Or programs use "unsafe" extensions
- * What should we call these issues?

"Memory Trespass"

- * "Memory trespass vulnerabilities are software weaknesses that allow memory accesses outside of the semantics of the programming language in which the software was written."
 - * Dai Zovi, "Security Applications of Dynamic Binary Translation", University of New Mexico Tech Report TR-CS-2002-38
 - * Yes, I am quoting myself. Deal with it.
- * Code injection and execution is only one way to exploit a few specific classes of memory trespass vulnerabilities

Or...

- * Type Violation
- * Type Safety Bypass
- * Memory Safety Bypass
- * Just don't say "buffer overflow" when it isn't
- * Don't get me started on the word "shellcode"



CONCLUSION

Conclusions

- * The threat landscape has changed in 2010, we can't just keep doing the same things we did in the 2000's
 - * The word "threatscape" is like "manscape" -- it should not be used in polite company
- * Vulnerabilities may be accidental, but attacks are not. Stop treating them like they are accidents.
- * Draw your own