A Brief Practical Security "Punch List"

Spring 2006 Internet2 Member Meeting --
Security Tools and the Campus Infrastructure
Security Survey Results Session

April 25, 2006, 8:45 AM-10:00 AM

Joe St Sauver, Ph.D. (joe@uoregon.edu)

University of Oregon Computing Center

http://www.uoregon.edu/~joe/punchlist/

Our Topics For This Part of Today's Session

- I've got about twenty minutes for my part of today's session, so let's focus on two big topics and two less complex ones if we have time:

  1. **Spoofed traffic and denial of service attacks**
  2. **Open recursive DNS servers and DNS amplification attacks**
  3. SpamAssassin and SURBLs
  4. Spammed blogs, wikis, guestbooks and other anonymously writable web pages

- Before we jump in, a couple of quick notes…

Talk Format and a Disclaimer

- These slides are quite detailed. Why?
  -- Time is limited and I'm prone to getting side tracked if I don't "stick to the script"
  -- I usually cover quite a bit of material fairly quickly
  -- I like to provide pointers to sources for further information (but hate to make you scribble URLs)
  -- I know these slides may be viewed after the fact by those who are not here today, and also by those whose primary language isn't English
  -- Some in the audience may be hearing impaired; think of these notes as closed captioning for them
- Disclaimer: all opinions expressed in this talk are my own, and do not represent the official position of UO, the Oregon Gigapop, I2, or any other entity.

I. Spoofed Traffic and Denial of Service Attacks

Please check and confirm that you are configured to prevent spoofed traffic from leaving your network.

Distributed Denial of Service (DDoS) Attacks

- As discussed in my May 3, 2005 Internet2 Member Meeting talk, "Explaining Distributed Denial of Service Attacks to Campus Leaders,"[1] in a distributed denial of service (DDoS) attack network traffic from thousands of hacked computer systems -- often systems located all over the Internet -- gets used in a coordinated way to overwhelm a targeted network or computer, thereby preventing the target from doing its normal work.
- Unlike that earlier general talk, today we **do** need to talk a little about a specific technical vulnerability. We need some quick background, first.

1) http://www.uoregon.edu/~joe/ddos-exec/ddos-exec.pdf

TCP and UDP Traffic

- There are basically two types of network application traffic: TCP and UDP.
- TCP traffic is associated with relatively persistent connections (such as ssh sessions, web traffic, email, etc.), and has a variety of characteristics which are desirable from a network application programmer's point of view, including retrans-mission of lost packets, congestion control, etc.
- UDP traffic, on the other hand, is designed for "send-it-and-forget-it" applications where you don't want to/can't afford to maintain state or you don't want a lot of connection setup overhead. DNS, NFS, and IP video traffic all normally run as UDP.

The Spoofability of UDP Connections

- Unlike a fully established TCP connection (which only gets established after a bidirectional handshake is negotiated and which is therefore robust to spoofing attempts),[2] UDP traffic can be created with virtually **any** apparent source address -- including IP addresses which have no relationship to the traffic's actual origin.
- Network traffic that's intentionally created with a bogus source address is said to be "spoofed."
- If allowed to reach the global Internet, spoofed traffic is generally indistinguishable from legitimate traffic.

2) Yes, of course, naked TCP SYNs are also spoofable.

Why Would Anyone Bother to Spoof Traffic?

- If you don't spend time "thinking like an attacker," you might not immediately "get" why an attacker would be interested in spoofing his attack traffic. The answer is actually quite simple: the attacker wants the systems he's using as part of his attack to stay online and unblocked as long as possible.
- Spoofing the source of the attack traffic…
  -- hinders backtracking/identification/cleanup of the system that's sourcing the traffic; and
  -- makes it harder for the attack victim to filter the attack traffic (the spoofed source addresses may be constantly changed by the attacker, and thus doesn't provide a stable "filterable characteristic").

"So Why Not Just Block All UDP Traffic?"

- Given that UDP can be easily spoofed by the bad guys/bad gals, sometimes you'll hear folks naively propose simply blocking all inbound or outbound UDP traffic (or at least heavily rate limiting all UDP traffic).
- Unfortunately, because some pretty basic services (like DNS) requires support for UDP, blocking (or heavily rate limiting) all inbound or outbound UDP traffic is generally **not** a good idea. :-; Warts and all, you have no choice but to learn to to live with UDP traffic. :-;

"Well, Can We Block SOME UDP Traffic?"

- For once, the answer is positive: yes, you can block some UDP traffic.
- For example, if you're the University of Oregon and your school has been assigned the IP address range 128.223.0.0-128.223.255.255 there's no reason for systems on your network to be sourcing packets that pretend to be from some other IP address range. We'd filter that spoofed traffic before it leaves our campus.
- This is a pretty basic sanity check, but you'd be surprised how many sites don't bother with even this trivial sort of filter.

Subnet-Level Filtering

- While it is great to prevent spoofing at the university-wide level, that sort of border router anti-spoofing filter does not prevent a miscreant from forging an IP address taken from one of your subnets for use on another of your subnets.
- *Cue subnet-level anti-spoofing filters….*
  You KNOW that hosts on each subnet should ONLY be originating packets with IP addresses legitimately assigned to that subnet, so at the uplink from each subnet, drop/block outbound packets that appear to be "from" any other IP address – another very basic sanity check.

Filtering at Other Levels of Granularity

- Although we've talked about filtering at your border and at each subnet uplink, you could also filter all the way upstream at the gigapop level, or all the way downstream at the host level.
- Obviously, the closer you get to the traffic source, the more effective the filter will be. That said, catching at least some problematic traffic at the gigapop level is better than nothing if you can't get your downstream customers to do the right thing closer to the traffic source (but the larger your gigapop, the harder it will be to keep accurate track of all the prefixes in use).

BCP38/RFC2827

- Let me be clear that ingress filtering of traffic with spoofed IP addresses is not new and is not my idea – it is Best Current Practice (BCP) 38/RFC 2827[3], written by Ferguson and Senie in May 2000.
- Unfortunately, despite being roughly six years old, **many** sites still do **NOT** do BCP38 filtering -- perhaps as many as 20-25% Internet wide.[4]
- **Does YOUR university do BCP38 filtering?**

3) http://www.ietf.org/rfc2827.txt

4) http://spoofer.csail.mit.edu/summary.php

"So Why Doesn't Everyone Do BCP38 Filtering?"

- "Too hard given the complexity of my network"
- Asymmetric costs/benefits: filtering my network protects you (which is nice), but filtering that traffic "costs" me w/o any tangible/economic "benefits." So what are these horrible "costs?"
  -- engineer time to configure and maintain the filters (one time/negligible for most .edu networks)
  -- overhead on the routers (but if that overhead is material enough to be a "show stopper," you should be upgrading anyway)
- "Too busy" (or other excuses)

"What's It To You Anyhow, Bub? Butt Out…"

- Some may question why others should care what they do with their networks – your network, your rules, right? Well, generally yes.
- However in this case, remember that if you're NOT doing BCP38 filtering, your network may be getting used to generate spoofed attack traffic that's pretending to be "from" someone else's network, and that's the point at which what you do (or don't do) potentially affects a lot of other people including the attack target itself, the entity whose IP addresses are being spoofed, etc.]

"So How Should I Be Doing This Filtering?"

- Only you can make the final decision about the best approach for your network, but you may want to see BCP84/RFC3704, March 2004.
- I would note, however, that strict mode unicast reverse path forwarding ("strict uRPF") is **not** a good idea for the multihomed environment typical of I2 universities due to route asymmetry.
- I would also urge you to review (April 19, 2006) draft-savola-bcp84-urpf-experiences-00.txt
- Quoting RFC3704 "Ingress Access Lists require typically manual maintenance, but are the most bulletproof when done properly…"

2. Open Recursive DNS Servers and DNS Amplification Attacks

Please make sure your name servers aren't answering recursive queries for random domains for random users.

A Specific Example of UDP Spoofing…

- Since we just got done covering UDP spoofing, talking a little about open recursive domain name servers and DNS amplification attacks seems like a "nice" segue/practical example of why BCP38 filtering is important, while also pointing out another specific vulnerability you should be addressing.
- Again, let's begin with a little background, first.

Thinking A Little About DNS

- Most users never really think about how DNS works[5] -- they just take it for granted that entering http://www.uoregon.edu/ in their web browser will take them to the University of Oregon home page. In order for that to happen, however, the web browser needs to be able to find out that www.uoregon.edu resolves to the IP address (or "dotted quad") 128.223.142.13
- The web browser, and ultimately the user, relies on the domain name system to do that name-to-dotted quad translation.
- DNS is thus a critical network service.

5) Geeks see RFC1035

Authoritative and Recursive DNS Servers

- There are different types of name servers, with "authoritative" and "recursive" DNS servers being the two most important types:

  -- Authoritative servers are definitive for particular domains, and should provides information about those domains (and ONLY those domains) to anyone.

  -- Recursive servers are customer-facing name servers that should answer DNS queries for customers (and ONLY for customers) concerning any domain.

- DNS servers that aren't appropriately limited can become abused.

For Example…

- Consider a situation where a DNS server is recursive AND is open for use by anyone (a server that's cleverly termed an "open recursive DNS server").
- While it might seem sort of "neighborly" to share your name server with others, in fact it is a really bad idea (the domain name system equivalent of running an open/abusable SMTP relay, in fact).
- The problem? Well, there are actually **multiple** problems, but one of the most important ones is associated with spoofed UDP traffic (see how this all ties together? :-;)

Spoofed DNS Attack Scenario

Dramatis personae:

- Attacker, who's working from non-BCP38 filtered network. Let's call him/her "A"
- Attack target – let's refer to that entity as "T"
- Open recursive domain name server on large, high bandwidth pipe, denoted below as "NS"

Act 1, Scene 1:

- "A" generates spoofed DNS queries with "T"'s address as the "source" address of the queries
- "NS" receives the spoofed queries and dutifully returns the "responses" for those queries to "T"
- "A" repeats as desired, thereby DoS'ing "T" via "NS"

Some Spoofed DNS Attack Scenario Notes

- -- From "T"'s point of view, the attack comes from "NS" not from "A"
  -- DNS queries are small and use UDP, so an attacker can generate a "large" query volume
  -- DNS response traffic is also UDP, which means that it is insensitive to net congestion.
  -- DNS responses can be **large** relative to size of DNS queries (output/input ratios can run over 8X on most DNS servers, and on servers supporting RFC2671 EDNS0 extensions, observed amplification can exceed 70X).
  -- "A" can employ **multiple query sources**, and **use multiple NS's** for more traffic (oh boy!)

This Is A Well Known Vulnerability

- I'm not letting the "cat out of the bag" about a big secret; this is a well known/documented threat:
  -- "The Continuing Denial of Service Threat Posed by DNS Recursion"[6]
  -- "DNS Amplification Attacks"[7]
  -- "DNS Distributed Denial of Service (DDoS) Attacks"

6) http://www.us-cert.gov/reading_room/DNS-recursion121605.pdf

7) http://www.isotf.org/news/DNS-Amplification-Attacks.pdf

8) www.icann.org/committees/security/dns-ddos-advisory-31mar06.pdf

Open Domain Name Servers Worldwide

- Unfortunately, despite this being a well known problem, it is estimated that 75% of all name servers worldwide run as open recursive name servers.[9]
- And in a spirit of self-criticism, feel free to note that UO's name servers were open until we secured them this past February 1st, 2006.[10]
- **If *our* domain name servers were open recursive until Feb 2006, *how about yours*? You NEED to get them secured.**

9) http://dns.measurement-factory.com/surveys/sum1.html

10) http://cc.uoregon.edu/cnews/winter2006/recursive.htm

The Problem Isn't "Just" About DDoS, Either

- By the way, if you aren't yet sufficiently motivated to "bite the bullet" and fix your DDoS-exploitable domain name servers, let me add a little more thrust to help launch that hog: if you're not controlling access to your domain name servers, you may also be leaving yourself vulnerable to **DNS cache poisoning attacks**, whereby vulnerable caching name servers can be made to return bogus results for a user's name service queries.[11]

11) http://www.lurhq.com/dnscache.pdf

What's a Cache Poisoning Attack?

- In a nutshell, in cache poisoning attacks, the attacker "primes" the caching name server to respond to queries with an IP address of his/her choice, rather than the real/normal IP address for that site. The innocent victim asks the caching name server for the IP address of a site of interest, such as the IP address of their bank's website. If the domain name of that site happens to be one that the attacker has poisoned, the victim is automatically and transparently misdirected to a website of the attacker's choice, rather than to their bank's real web site, and confidential data can then end up being lost.

Another Cache Poisoning Scenario

- Another cache poisoning scenario uses cache poisoning to redirect queries for popular sites (such as google.com or hotmail.com) to a site that contains a virus or other malware. If your caching name server has been poisoned, when you try to visit one of these popular sites, you can unknowingly be redirected to another site that stealthily tries to infect your PC with malware.
- Blocking open access to your recursive name servers won't completely eliminate the possibility of your servers participating in such attacks, but it will reduce the likelihood of that sort of abuse.

Recommendations

- Insure that you're running a current version of BIND[12] (or whatever DNS software you use)
- Insure that you've separated your Internet-facing authoritative name server from your customer-facing recursive name server
- Protect your customer-facing recursive name server from access by non-customers
- Consider analyzing DNS traffic with DNStop[13]
- Consider donating DNS log data to the RUS-CERT Passive DNS Replication Project[14]

12) http://www.isc.org/index.pl?/sw/bind

13) http://dns.measurement-factory.com/tools/dnstop/

14) http://cert.uni-stuttgart.de/stats/dns-replication.php

3. SpamAssassin and SURBLs

Speaking of spam, if you're not using SpamAssassin with SURBL support, you're seeing a lot more email spam than you really need to…

Spamhaus' Effective Spam Filtering Recommendations

- We've traditionally relied on source-based spam filtering (blocking known spammer infested domains and network blocks, compromised spam zombied systems, etc.) More recently, however, we've begun using a two stage approach, first doing traditional connect-time source-based filtering with the Spamhaus SBL+XBL and other filters, and then checking the remaining messages for spammer URLs in the message body as recommended by Spamhaus.[15]

15) http://www.spamhaus.org/effective_filtering.html

The Magic of the SURBL

- Those of you who know me well know that I've traditionally been leary of content-based spam filtering approaches, but URIBLs (lists of spamvertised web sites, see for example http://www.surbl.org/) have truly changed my mind, and UO's now routinely tagging all incoming uoregon.edu mail with SpamAssassin scores[16]
- Nothing visible happens to mail based on those scores unless users opt-in via a web interface (https://password.uoregon.edu/spam/). If they do opt-in, there's not much spam left after that… :-)

16) http://cc.uoregon.edu/cnews/spring2006/spamassassin.htm

SURBL Test Scores in SpamAssassin 3.1[17]

- A measure of the value of the SURBL tests…

```
URIBL_SC_SURBL 3.600 (SpamCop)
URIBL_JP_SURBL 3.360 (Joe Wein+Prolocation)
URIBL_AB_SURBL 3.306 (AbuseButler)
URIBL_OB_SURBL 2.617 (Outblaze)
URIBL_PH_SURBL 2.240 (Phishing)
URIBL_WS_SURBL 1.533 (Bill Stearns)
```

  Given that a score of 5.0 is typically sufficient for a message to be tagged or foldered as spam, clearly these are powerful tests.

- For comparison for those of you familiar with the traditional Spamhaus SBL and XBL DNSBLs:

```
RCVD_IN_XBL 3.114
RCVD_IN_SBL 2.712
```

17) http://spamassassin.apache.org/tests_3_1_x.html

Recommendation

- I'll give y'all the same recommendation I made to the carrier Messaging Anti-Abuse Working Group (MAAWG) during their 6th meeting last month in San Francisco:[18] if you're not using the SURBL, you should be. For Internet2 universities, SpamAssassin 3.1.1 is one easy way to do so.
- Oh yes: you'll probably want to arrange for rsync access to the SURBL.[19]

18) http://www.uoregon.edu/~joe/maawg6/maawg-sfo.pdf

19) http://www.surbl.org/rsync-signup.html

4. Spammed Blogs, Wikis, Guestbooks and Other Anonymously Writable Web Pages (and Archived Unfiltered Mailing Lists)

Bet you don't know what your college web pages are being used to advertise…

The King of the Internet Applications and Its Latest Manifestations

- The world wide web is arguable the king of all Internet applications (realistically challenged only by email IMHO).
- Lately, much of the web-related buzz has been about blogs (web logs), wikis (collaborative document development spaces), and other online web sites where users can react to what they read or contribute to the refinement of the content on those sites.

But **The Roaches of the Internet Have Found Your Users' Blogs, Wikis and Guestbooks**

- Given the exuberance with which spammers have attacked e-mail, Usenet, instant messaging and other online fora, it should come as no surprise that spammers are also hard at work spamming academic blogs, wikis, guestbooks and other anonymously writable web sites with ads for mail-order controlled substances, online casinos, porn, 'free* 42" color TVs,' pirated software, discount mortgage leads, penny stocks, etc.
- These spammers hope to either improve their page rank in Google, or to directly attract customers to the web sites they're spamming, or both.

Surprise Yourself

- I'd originally thought about sharing per-institution results a variety of commonly spamvertised query search terms for a representative sampling of Internet2 universities, but in the interest of time and to avoid blushes, I'll leave that as a "homework project" for the curious. If you are interested in checking, pick a commonly spamvertised service or product, and restrict your search to a site of interest. E.G., Google for something like: phentermine site:sample.edu
- Things to note: some references may be legitimate (non spammy), while other references may be automatically suppressed by Google.

Lots of Abuse, But That's Nothing New

- Anonymously writable online resources have always been vulnerable to abuse: remember the old days when people would have anonymously writable ftp sites until the d00dz began to use them as drop sites for warez, stolen credit cards, and similar things? These days it is rare to see someone offering an anonymously writable ftp site (or at least you don't seem them offering it for long!).
- Fortunately less extreme countermeasures are available for your institution's blogs, wikis, guestbooks, and mailing list archives.

Some Steps You May Want to Consider Taking

- Consider moderating all comments
- Consider requiring commenting users to auth
- Upgrade to the latest version of the blog/wiki software you're using; in many cases, current versions of the software that may be currently getting abused on your system will include good integrated anti-comment-spam functionality
- Be aware that some sites have had success with blocklist based comment filters; see, for example: http://meta.wikimedia.org/wiki/Spam_blacklist
- One popular free-for-non-profit-entities "content evaluation" solution that works with a variety of platforms is Akismet (see http://akismet.com/)

Some Web Trivia Noted In Passing

- 4,236 American colleges and universities
- 2.67 billion .edu web pages known to Google (==> 630,311 pages/university if uniformly spread)
- 4 universities (MIT, Berkeley, Stanford, Harvard) collectively account for over 1/4th of that total; MIT alone has a staggering 278 million pages
- 22 universities (4 schools already mentioned plus Washington, Wisconsin, Texas, Illinois, Cornell, Michigan, Yale, Columbia, Penn State, UCLA, Chicago, Maryland, Penn, Vanderbilt, Virginia, Minnesota, Princeton and Indiana) collectively account for over 1/2 of all .edu pages

See: http://cc.uoregon.edu/cnews/spring2006/webrank.htm

Questions?

- Thanks for the chance to talk today!
  Are there any questions?