TCP/IP Security Holes: A Look Back

Steven M. Bellovin
smb@research.att.com
http://www.research.att.com/~smb
AT&T Labs Research

The Original Paper

- A protocol-level analysis of security issues
- Buggy code and misadministration issues not considered
- My first public foray into Internet security issues

Some Background

- I was one of the instigators of the Bell Labs intranet
- Things sometimes *broke*
- A number of incidents had me thinking about security (I caught my first hackers in 1971)
- I'd started working on firewalls — but what were the threats that the firewall should block?

What Broke?

- Often, there was a routing problem or an address assignment problem
- It was clear that if something could happen by accident, it could be done deliberately
- What were the implications?
- After thinking about it a lot, I decided to write it up
- Information from original paper in red

Sequence Number Attacks

- Invented by Morris (1984)
- Examine the TCP sequence number received for one connection; use that to predict the sequence number for another connection
- Use that to acknowledge packets never received, and thus impersonate another host
- Primary target: the *rsh* command, which relied on address-based authentication

Implications of Sequence Number Guessing

- Attack ignored for several years, until Mitnick used it against Shimomura
- I devised a fix in 1996, but it's often considered too expensive
- Random increments are worse than I had thought — statistical investigation shows that the increments tend to converge, thus making guessing more feasible
- Many other fixes break correctness properties of initial sequence number
- Combining sequence number guessing with large TCP window sizes can lead to TCP Reset attacks on BGP

Some Hosts Disclose Sequence Numbers

- When I wrote the paper, TOPS-20 hosts had a *netstat* service that disclosed sequence numbers of active connections
  (I intentionally omitted that detail from the paper)
- More recently, there have been proposals to include TCP sequence numbers in SNMP MIBs
- Sequence number guessing is a threat to many other protocols

Security Architecture and Sequence Number Guessing

- Address-based authentication was not part of the *standardized* protocols; it was used by Berkeley's *r*-commands
- Sequence numbers have no guaranteed security properties; one should not assume that they do
- Bottom line: TCP is not (and is not intended to be) a secure protocol; if you need security, it should be provided above or below the TCP layer
- Too many application designers have ignored that!

Routing Attacks

- Many forms of routing attack: source routing, fake RIP messages, EGP spoofing
- Several consequences given, including address-spoofing, connection hijacking, and eavesdropping
- Routing protocol attacks are difficult to defend against, because lies can be propagated by honest routers
- Firewalls can sometimes block address impersonation based on routing attacks

Routing Attacks Are Real

- Many routing incidents have happened on the backbone
- Example: “AS 7007 incident”, where a small ISP claimed that it had the best route to most of the Internet
- Spammers sometimes announce false routes, dump their garbage, withdraw the routes, and run
- Some evidence for more sophisticated routing attacks

Securing Routing

- Many sites use keyed MD5 authentication to protect BGP connections
- Major ISPs filter customer BGP announcements
- But hop-by-hop security isn't enough
- Fixes have been proposed, but not adopted
- No strong, standardized security mechanisms for OSPF or BGP
- Routing remains a major security vulnerability
- End-to-end encryption protects against many of the risks of routing attacks

ICMP Attacks

- Use ICMP Redirect for routing attacks
- Use ICMP error packets for denial of service
- ICMP Redirects were never very real as an attack mechanism
- The denial of service attacks did occur, though that trend died down as hosts became smarter

Identification Protocol

- Premise: ask the originating host who owns a given TCP connection
- Lots of risks, especially if the host lies
- Not a standards-track protocol at the time
- Modernized version *is* standards-track; often used for logging
- Just as useless
- My laptop replies with “ident-is-a-completely-pointless-protocol-that-offers-no-security-or-traceability-at-all-so-take-this-and-log-it!”

Email

- “It is quite vulnerable to misuse”
- “No authentication mechanisms”
- “Leaves the door wide open to faked messages”

Email

- No, I didn't predict spamming or phishing. . .
- Authenticated email now exists (even if it's little-used), but authentication doesn't stop spam
- It doesn't even do much to stop phishing: consider `paypa1.com` versus `paypal.com`

☞ Cryptographic authentication is useless unless users (or their mailers) check the signatures and certificates against *known-good* sources

A Digression into Human Factors

Compare:

`paypa1.com`
`paypal.com`

versus

paypa1.com
paypal.com

Fonts matter — in real life and on slides

The DNS

- Sequence number attacks are possible
- Enemies can abuse the DNS to redirect traffic to their hosts
- Possible to use the the DNS for data-gathering

DNS Problems Are Even Worse Than I Said. . .

- Sequence number attacks have been implemented
- Attacks on DNS registries (including social engineering attacks) have been used to corrupt the authoritative name servers
- DNS cache contamination attacks can plant fake data
- Possible to trick the *r*-commands via fake PTR records in enemy-controlled DNS zones (today, generally blocked by cross-checks)
- DNSsec — digitally-signed DNS records — is a strong defense, but it has been much harder to develop than thought, and is not yet deployed.

Reserved Ports

- Berkeley declared that ports < 1024 were “privileged”, and only available to `root`
- It was a bad idea then; it’s worse now, given the number of single-user machines on the Net
- It could have been far worse — FTP “bounce attacks” can send arbitrary data *from* port 21, in the restricted range; fortunately, the *r*-utilities only trust ports ≥ 512

Cryptography

- Cryptographic authentication is the *only* reliable way to do authentication on the Internet
- But it's *hard* to get right
- Cryptography is used much less than it should be, and its major use — SSL — is more of a fig leaf against a sophisticated attack
- But we don't have many sophisticated attackers, because the easy attacks are so successful. . .

Security Problems in Today's Internet

- Most system penetrations are due to buggy code or weak passwords, rather than protocol flaws
- That said, we have seen sequence number attacks, routing attacks, DNS attacks, etc., in the wild
- Attackers have shown an ability to exploit any weaknesses; the prevalence of code-based attacks is more due to their ease than to the strength of our protocols
- The most common protocol-level attack is connection-hijacking based on ARP-spoofing

Lessons for Protocol Design

- Protocols should be analyzed for security during development
- We have a decent grasp on security properties for client-server protocols; it's also relatively easy to bolt on crypto above or below most such protocols
- Caution: pay careful attention to interfaces and guarantees — that's where the holes creep in
- However, multi-party protocols — SIP, Diameter, various peer-to-peer protocols — are much harder to secure
- The difficult question is *authorization* to perform some activity

Have We Learned?

- Yes, though progress hasn't been as rapid as I would have liked
- Too many older RFCs say "Security issues are not discussed in this memo"
- Fortunately, that is unacceptable for today's protocol designs; RFCs are supposed to include suitable security mechanisms, thorough security analyses, and documentation of residual risks (RFC 3552)
- Some of the recommendations in the paper — cryptography, firewalls, intrusion detection — are used, though not always as much or as well as I'd like
- "You can lead a horse to water. . . "