Advanced Network Security

Richard Clayton

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Check Point Course

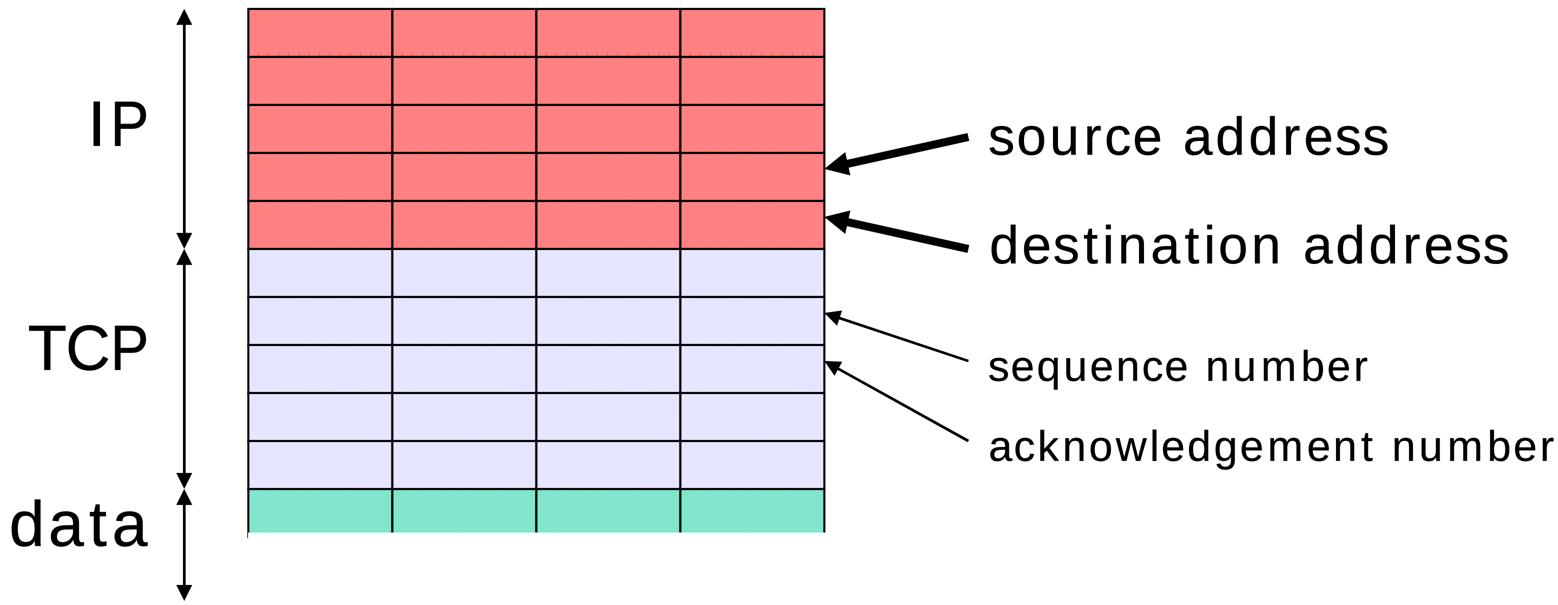
25 June 2010

Outline

- Traceability
  - and how edge devices fail to help
- Stealing service
  - and how edge devices make it easier
- Is the infrastructure secure ?
  - attacks on DNS
  - attacks on BGP

Traceability:
who did that?

(Almost) all you need to know about TCP/IP



Traceability

- Destination address is always valid
- Can send bad packets with 1-way traffic
- Source address is valid for 2-way traffic
- Locate ISP of sender by consulting RIR (RIPE, ARIN, APNIC, LACNIC, AfriNIC) whois records
- Ask ISP to reveal usage at the specific time
- Lots of assumptions underlie this process, but it’s usually accurate (if logs exist) – except at the very last stage, identifying the user

Spoofing

- 3-way handshake

-->	SYN	client offset
<--	SYN-ACK	server offset
-->	ACK	

- If offset (and other info) is predictable don’t need to see the return traffic to have a successful conversation
- Described by Morris (85) and CERT (95)
- Fix by making sequence numbers <u>**random**</u> and perhaps by suitable packet filtering at borders

Traceability fails at the egde

- Network Address Translation
  - may be part of a firewall, or router solution
  - used to preserve IP address space
  - used to hide network architecture
  - unlikely to be logged
- DHCP
  - dynamic allocation of addresses
  - logging can also be problematic
  - underlying assumption that MAC addresses constant

Mobile IP providers

- Data phones and Internet “dongles” for laptops mean millions of new TCP/IP users
- BUT providers cannot obtain huge blocks of IP address space (IPv4 will soon be exhausted)
- So they are using NAT, with many (hundreds) of users sharing the same IP address
- Hence need to provide IP address + timestamp (& timezone) PLUS port number
- Existing security logging often inadequate
- AND not addressed by Data Retention Directive

Stealing service

“Practical anonymity”

- Steal a password
- Use a free account and withhold your CLI
- Use a pre-paid WAP phone
- Use a cybercafé
- Use someone else’s WiFi
- Multiple jurisdictions will slow tracing down
  - though perhaps avoid the USA
- NB: Best Practice is far from universal
- or you could just go into work and use the LAN

Complex identity theft

- Borrow IP address <u>and</u> MAC address
  - if real owner isn’t present then will work just fine!
  - all the logs (if any) will point at them
- Investigators will have to resort to CCTV footage, building entry records or holes in the record of activity of your machine
- So wait until real owner is at their desk
  - sniff traffic (easy on WiFi, complex if switched)
  - their TCP/IP stack will notice unexpected packets
  - so need to do something about their TCP resets...

TCP resets

Start to talk to a mail server

```
1028 > smtp [SYN]       Seq=0 Ack=0 Win=32768 MSS=1460
smtp > 1028 [SYN, ACK] Seq=0 Ack=1 Win=17520 MSS=1460
```

But real owner of identity sends reset to the mail server

```
1028 > smtp [RST]       Seq=1 Ack=4087568586 Win=0
```

So when we do third packet of handshake we are rebuffed

```
1028 > smtp [ACK]       Seq=1 Ack=1 Win=32768
smtp > 1028 [RST]       Seq=1 Ack=207398712 Win=0
```

Software firewalls

- In 2004 built a rule-breaking ethernet interface that collided with unwanted RSTs
- Device worked, so put into PhD thesis in 2005
- Encountered an unexpected difficulty generating dumps of RST packets for thesis chapter
- Eventually found that “ZoneAlarm” was discarding incoming SYN/ACK (and other segments) for an unknown connection
  - TCP/IP stack didn’t see packets so no RSTs generated!
- Microsoft XP firewall does the same

Stealth mode: an urban myth

- Bastion firewalls try and hide machines
  - slow down the hackers by obscuring detail
- Copied by “software firewalls”
  - despite them serving a different purpose
- Shields Up! made “stealth mode” a virtue
  - assumes that attackers probe and then pounce
  - assumes attackers are single threaded

Wireless hotspots

- Airports (etc) charge for wireless access
- Hence can borrow the identity of nearby Windows XP (etc) user – whose firewall is almost certainly enabled "to be safe"
- Airport could (probably) spot the subterfuge by analysis of port number usage etc
  - cf: counting hosts behind a NAT
- Economic analysis interesting : no incentive on software firewall maker to develop a fix

Robert in India

- Could see backbone wireless AP but not those meant to be used by customers
- Spoofed the IP address and MAC of an AP
- Identified gateway address (eventually)
- Ensured did not send RSTs or ICMPs

```
net.inet.tcp.blackhole = 2
net.inet.udp.blackhole = 1
```

- Bob’s your uncle! ☺

Take homes

- TCP must use truly random initial values to avoid spoofing
- Ethernet addressing works through convention and cooperation
- Switched networks reduce opportunities for identity theft – but 802.11 WiFi can bring them right back again
- Firewalls don’t always make you safer!

All your mailserver are belong to us

Threat scenario

- I wish to capture a significant amount of incoming email to a major ISP mail server
  - email may contain passwords etc
  - email can be made to contain passwords etc
  - answering email often “proves” identity
  - obvious opportunity to blackmail the ISP, or just trash their reputation as being secure
- Attack should “scale” to many ISPs
  - 0-day exploit on `sendmail` not considered here

Resources

- Back bedroom attackers
  - can now have control of a reasonable size botnet
- Criminal entrepreneurs
  - may own (or 0wn!) a smallish ISP in Ruritania
- Organised crime ??
  - simpler for them just to bribe an employee!
- I am NOT assuming that BGP or DNS are too obscure to be attacked effectively

Underlying strategies

- Cannot just steal packets – people notice
  - cf YouTube outage in February 2008 (Pakistan Telecom)
- Accept email, resend to the correct ISP
  - top 50 senders is a give-away, so use botnet
- Reject email end of data with a 4xx response
  - email generally re-delivered after a delay, so suitable for intermittent attacks
- Tunnel SMTP packets to correct place
  - either a peer of target or customer within target

DNS (I): active attacks

- DNS server asks for data
  - checks answer has correct identifier field
  - attacker supplies incorrect answer first
    - 16 bit identifier is not long enough!
    - hence modern software randomises request port
- Older software is flawed
  - predictable random numbers!
    - or even accepts non-authorised data!
- No-one monitors for attacks
  - however this scales badly, so of limited interest
  - BUT WAIT!

DNS (II): Kaminsky

- Ask for multiple sub-domains (sub1, sub2 etc.)
  - neat way of ensuring resolver always has to ask
- Attacker tries to get their answer in first
  - BUT of course only poisons some obscure sub-domain
- Kaminsky realised could supply NS data as well
  - "in-bailiwick" data (extra info from authoritative server)
  - relied upon for some purposes! So devastating attack!
- Mitigate (only) with lots of entropy (as before)
  - and what of clever servers behind dumb firewalls?
  - only real fix is DNSSEC

DNS (III): phishing

- “Rock-phish” gang spoofed GoDaddy Aug07
  - probably just wanted some cheap domains
  - BUT control of a registrar account permits changes to name server identities
- Registrars for grown-ups will check validity of changes out-of-band, $10 hosting will not
  - significant number of US banks were vulnerable
- Attack vector might also be malware...

DNS (IV): root of trust

- 13 top level name servers (A-M)
  - maximum that will fit in a DNS response
- Included with BIND (etc) as a text file
  - you have to start bootstrapping somewhere!
- L moved from 198.32.64.12 to 199.7.83.42
  - moved 1 Nov 2007 (warnings sent 24 Oct 2007)
  - AS20144 (ICANN) announced route until 2 May 2008
- BUT other AS's announced route in 2008/9
  - Dec 15 (AS42909), Mar 18 (AS 4555), Apr 1 (AS9584)
  - all serving the right thing (through May, we think!)

Attacks on BGP

- Basic idea: announce a /32 for mailserver
  - BGP prefers a “more specific” announcement
- Traffic then flows to Ruritania
  - email contents are available for inspection
- /32 may not propagate, so /24 may be better
  - leads to complexity if other hosts or services on /24
  - hence tunnelling packets back to ISP may be best (and just sniff them as they pass)
- Sniffing possible anyway at other ISPs
  - difference here is scale and remoteness

More specifics...

- Route should not be accepted
  - mnt-lower prevents creation of new route objects
  - so everyone ought to notice that route isn't valid
  - complexities with multiple route registries
- Route may be spotted by monitoring
  - MyASN @ RIPE, Renesys & some academic projects

    `http://iar.cs.unm.edu/alerts.php`

    `http://phas.netsec.colostate.edu`
  - note that bogon filtering hides route from owner! and so Best Practice prevents give-away failures

Unauthorised announcements

- Existing route: hope to be a shorter AS path
  - BGP counts AS's to determine preference
  - so more effective in Ruritania than London
- May help to forge origin for peer to accept the route (entirely dependent on filters)
- Once again, monitoring detects wickedness
  - but registry data error-prone and incomplete so can perhaps only consider changes?
  - and of course you need to know all about multi-homed customers! Is this possible?

SMTP Defence I: encryption

- Opportunistic encryption (RFC3207)
  - uses STARTTLS capability & command
  - negotiate mutually acceptable algorithm
- Plus points:
  - works out of the box for major MTAs
  - only end-points can decrypt the traffic
- Minus points:
  - increases processing load (may not matter)
  - no “man-in-the-middle” protection

SMTP Defence II: authentication

- Check certificates before sending email
  - prevents man-in-the-middle
- Plus points:
  - works out of the box for major MTAs
- Minus points:
  - increases processing load (albeit may not matter)
  - needs a Public Key Infrastructure (or a lot of bilateral arrangements), so perhaps store in DNS?

Network level defences

- Anti-spoofing filters on customer links
  - motherhood! (but tedious for custom customers)
- Much harder to do on border routers
  - unicast reverse path forwarding (RPF) can help
  - but at IXPs this may not be practicable
- Can check if traffic coming from correct peer
  - straightforward(ish) sFlow/Netflow analysis

Secure DNS/BGP

- Secure DNS almost here
  – some TLDs already signed, more to come
  – unlikely that will be fully deployed for years
  – BUT Kaminsky exploit has given it a huge boost

- Secure BGP(s) experimental at present
  – concerns about performance (cf MD5)
  – concerns about key distribution
  – when will it be stable and inter-working?

Blended attacks

- Some key distribution schemes use DNS
- Attack the DNS and you may be able to compromise systems that are “secure”
- Best use of a BGP attack may be to capture the DNS servers (think long TTL), and then you can go after the mail servers at leisure!
- ...and of course you may just want to DoS
  - so you don’t mind if your attack is noticed

But why not just attack the customer directly?

Customer equipment

- Windows machines may keep name server identities in registry – easy for malware to change
- But in practice, usually set by DHCP
- Hence only need to compromise home routers
  - may have no password at all (and insecure wireless)
  - may be configurable from “the outside”
  - may be insecure, with buffer overflows &c
  - may still have the standard password
- With wireless as well, some researchers postulate an out-of-band worm!

Negligence

- The failure to use reasonable care
- Current test for “duty of care”:
  - harm must be (1) reasonably foreseeable
    (2) there must be a relationship of proximity between the plaintiff and defendant and
    (3) it must be “fair, just and reasonable” to impose liability
- If one of my attacks is effective on a mailserver, because of firewall failings, are you negligent?
- Short term specific: if your router/firewall makes DNS IP-IDs predictable, are you negligent?

Advanced Network Security

Richard Clayton

http://www.lightbluetouchpaper.org

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Check Point Course

25 June 2010