CSci 5271
Introduction to Computer Security
DoS, Tor, and usability combined slides

Stephen McCamant
University of Minnesota, Computer Science & Engineering

Preview question

Officially the name of the Tor network is not an acronym, but the "or" part of the name originated from this technique it uses:

A. onion routing
B. oatmeal reciprocity
C. one-time resilience
D. oilseed relaying
E. oblivious ratcheting

Outline

Denial of service and the network (cont'd)
Anonymous communications techniques
Announcements intermission
Tor basics
Tor experiences and challenges
Usability and security
Usable security example areas

DoS versus other vulnerabilities

- Effect: normal operations merely become impossible
- Software example: crash as opposed to code injection
- Less power that complete compromise, but practical severity can vary widely
  - Airplane control DoS, etc.

Compression DoS

- Some formats allow very high compression ratios
  - Simple attack: compress very large input
- More powerful: nested archives
- Also possible: "zip file quine" decompresses to itself

DoS against network services

- Common example: keep legitimate users from viewing a web site
- Easy case: pre-forked server supports 100 simultaneous connections
- Fill them with very very slow downloads

Tiny bit of queueing theory

- Mathematical theory of waiting in line
- Simple case: random arrival, sequential fixed-time service
  - M/D/1
- If arrival rate $\geq$ service rate, expected queue length grows without bound

SYN flooding

- SYN is first of three packets to set up new connection
- Traditional implementation allocates space for control data
- However much you allow, attacker fills with unfinished connections
- Early limits were very low (10-100)

SYN cookies

- Change server behavior to stateless approach
- Embed small amount of needed information in fields that will be echoed in third packet
  - MAC-like construction
- Other disadvantages, so usual implementations used only under attack

DoS against network links

- Try to use all available bandwidth, crowd out real traffic
- Brute force but still potentially effective
- Baseline attacker power measured by packet sending rate

Traffic multipliers

- Third party networks (not attacker or victim)
- One input packet causes n output packets
- Commonly, victim's address is forged source, multiply replies
- Misuse of debugging features

"Smurf" broadcast ping

- ICMP echo request with forged source
- Sent to a network broadcast address
- Every recipient sends reply
- Now mostly fixed by disabling this feature

Distributed DoS

- Many attacker machines, one victim
- Easy if you own a botnet
- Impractical to stop bots one-by-one
- May prefer legitimate-looking traffic over weird attacks
  - Main consideration is difficulty to filter

Outline

Denial of service and the network (cont'd)

Anonymous communications techniques

Announcements intermission

Tor basics

Tor experiences and challenges

Usability and security

Usable security example areas

Traffic analysis

- What can you learn from encrypted data? A lot
- Content size, timing
- Who's talking to who
  - → countermeasure: anonymity

Nymity slider (Goldberg)

- Verinymity
  - Social security number
- Persistent pseudonymity
  - Pen name ("George Eliot"), "moot"
- Linkable anonymity
  - Frequent-shopper card
- Unlinkable anonymity
  - (Idealized) cash payments

Nymity ratchet?

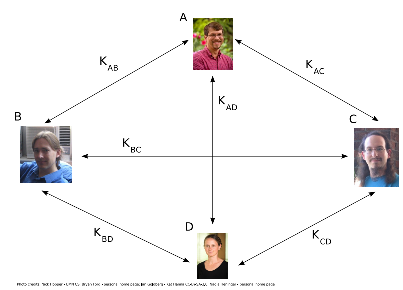
- It's easy to add names on top of an anonymous protocol
- The opposite direction is harder
- But, we're stuck with the Internet as is
- So, add anonymity to conceal underlying identities

Steganography

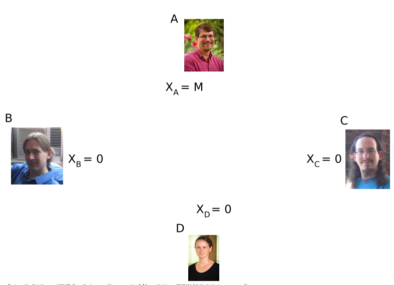
- One approach: hide real content within bland-looking cover traffic
- Classic: hide data in least-significant bits of images
- Easy to fool casual inspection, hard if adversary knows the scheme

Dining cryptographers

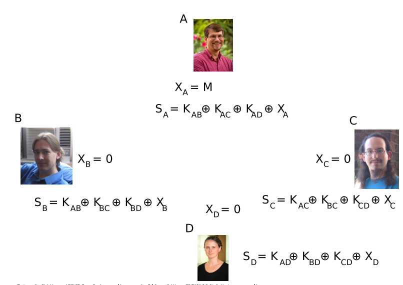
Dining cryptographers



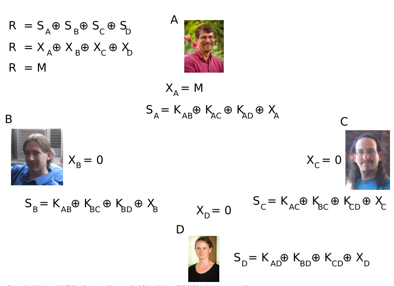
Dining cryptographers



Dining cryptographers



Dining cryptographers



DC-net challenges

- Quadratic key setups and message exchanges per round
- Scheduling who talks when
- One traitor can anonymously sabotage
- Improvements subject of ongoing research

Mixing/shuffling

- Computer analogue of shaking a ballot box, etc.
- Reorder encrypted messages by a random permutation
- Building block in larger protocols
- Distributed and verifiable variants possible as well

Anonymous remailers

- Anonymizing intermediaries for email
  - First cuts had single points of failure
- Mix and forward messages after receiving a sufficiently-large batch
- Chain together mixes with multiple layers of encryption
- Fancy systems didn't get critical mass of users

Outline

Announcements: this week

- Next and final progress reports due Wednesday night
- Wednesday lecture will be electronic cash and blockchains only

Outline

Tor: an overlay network

- Tor (originally from "the onion router")
  - `https://www.torproject.org/`
- An anonymous network built on top of the non-anonymous Internet
- Designed to support a wide variety of anonymity use cases

Low-latency TCP applications

- Tor works by proxying TCP streams
  - (And DNS lookups)
- Focuses on achieving interactive latency
  - WWW, but potentially also chat, SSH, etc.
  - Anonymity tradeoffs compared to remailers

Tor Onion routing

- Stream from sender to D forwarded via A, B, and C
  - One Tor circuit made of four TCP hops
- Encrypt packets (512-byte "cells") as $E_A(B, E_B(C, E_C(D, P)))$
- TLS-like hybrid encryption with "telescoping" path setup

Client perspective

- Install Tor client running in background
- Configure browser to use Tor as proxy
  - Or complete Tor+Proxy+Browser bundle
- Browse web as normal, but a lot slower
  - Also, sometimes `google.com` is in Swedish

Entry/guard relays

- "Entry node": first relay on path
- Entry knows the client's identity, so particularly sensitive
  - Many attacks possible if one adversary controls entry and exit
- Choose a small random set of "guards" as only entries to use
  - Rotate slowly or if necessary
- For repeat users, better than random each time

Exit relays

- Forwards traffic to/from non-Tor destination
- Focal point for anti-abuse policies
  - E.g., no exits will forward for port 25 (email sending)
- Can see plaintext traffic, so danger of sniffing, MITM, etc.

Centralized directory

- How to find relays in the first place?
- Straightforward current approach: central directory servers
- Relay information includes bandwidth, exit polices, public keys, etc.
- Replicated, but potential bottleneck for scalability and blocking

Outline

Denial of service and the network (cont'd)
Anonymous communications techniques
Announcements intermission
Tor basics
Tor experiences and challenges
Usability and security
Usable security example areas

Anonymity loves company

- Diverse user pool needed for anonymity to be meaningful
  - Hypothetical Department of Defense Anonymity Network
- Tor aims to be helpful to a broad range of (sympathetic sounding) potential users

Who (arguably) needs Tor?

- Consumers concerned about web tracking
- Businesses doing research on the competition
- Citizens of countries with Internet censorship
- Reporters protecting their sources
- Law enforcement investigating targets

Tor and the US government

- Onion routing research started with the US Navy
- Academic research still supported by NSF
- Anti-censorship work supported by the State Department
  - Same branch as Voice of America
- But also targeted by the NSA
  - Per Snowden, so far only limited success

Volunteer relays

- Tor relays are run basically by volunteers
  - Most are idealistic
  - A few have been less-ethical researchers, or GCHQ
- Never enough, or enough bandwidth
- P2P-style mandatory participation?
  - Unworkable/undesirable
- Various other kinds of incentives explored

Performance

- Increased latency from long paths
- Bandwidth limited by relays
- Recently 1-2 sec for 50KB, 3-7 sec for 1MB
- Historically worse for many periods
  - Flooding (guessed botnet) fall 2013

Anti-censorship

- As a web proxy, Tor is useful for getting around blocking
- Unless Tor itself is blocked, as it often is
- *Bridges* are special less-public entry points
- Also, protocol obfuscation arms race (uneven)

Hidden services

- Tor can be used by servers as well as clients
- Identified by cryptographic key, use special rendezvous protocol
- Servers often present easier attack surface

Undesirable users

- P2P filesharing
  - Discouraged by Tor developers, to little effect
- Terrorists
  - At least the NSA thinks so
- Illicit e-commerce
  - "Silk Road" and its successors

Intersection attacks

- Suppose you use Tor to update a pseudonymous blog, reveal you live in Minneapolis
- Comcast can tell who in the city was sending to Tor at the moment you post an entry
  - Anonymity set of 1000 $\rightarrow$ reasonable protection
- But if you keep posting, adversary can keep narrowing down the set

Exit sniffing

- Easy mistake to make: log in to an HTTP web site over Tor
- A malicious exit node could now steal your password
- Another reason to always use HTTPS for logins

Browser bundle JS attack

- Tor's Browser Bundle disables many features try to stop tracking
- But, JavaScript defaults to on
  - Usability for non-expert users
  - Fingerprinting via NoScript settings
- Was incompatible with Firefox auto-updating
- Many Tor users de-anonymized in August 2013 by JS vulnerability patched in June

Traffic confirmation attacks

- If the same entity controls both guard and exit on a circuit, many attacks can link the two connections
  - "Traffic confirmation attack"
  - Can't directly compare payload data, since it is encrypted
- Standard approach: insert and observe delays
- Protocol bug until recently: covert channel in hidden service lookup

Hidden service traffic conf.

- Bug allowed signal to guard when user looked up a hidden service
  - Non-statistical traffic confirmation
- For 5 months in 2014, 115 guard nodes (about 6%) participated in this attack
  - Apparently researchers at CMU's SEI/CERT
- Beyond "research," they also gave/sold info. to the FBI
  - Apparently used in Silk Road 2.0 prosecution, etc.

Outline

Denial of service and the network (cont'd)
Anonymous communications techniques
Announcements intermission
Tor basics
Tor experiences and challenges
Usability and security
Usable security example areas

Users are not 'ideal components'

- Frustrates engineers: cannot give users instructions like a computer
  - Closest approximation: military
- Unrealistic expectations are bad for security

Most users are benign and sensible

- On the other hand, you can't just treat users as adversaries
  - Some level of trust is inevitable
  - Your institution is not a prison
- Also need to take advantage of user common sense and expertise
  - A resource you can't afford to pass up

Don't blame users

- "User error" can be the end of a discussion
- This is a poor excuse
- Almost any "user error" could be avoidable with better systems and procedures

Users as rational

- Economic perspective: users have goals and pursue them
  - They're just not necessarily aligned with security
- Ignoring a security practice can be rational if the rewards is greater than the risk

Perspectives from psychology

- Users become habituated to experiences and processes
  - Learn "skill" of clicking OK in dialog boxes
- Heuristic factors affect perception of risk
  - Level of control, salience of examples
- Social pressures can override security rules
  - "Social engineering" attacks

User attention is a resource

- Users have limited attention to devote to security
  - Exaggeration: treat as fixed
- If you waste attention on unimportant things, it won't be available when you need it
- Fable of the boy who cried wolf

Research: ecological validity

- User behavior with respect to security is hard to study
- Experimental settings are not like real situations
- Subjects often:
  - Have little really at stake
  - Expect experimenters will protect them
  - Do what seems socially acceptable
  - Do what they think the experimenters want

Research: deception and ethics

- Have to be very careful about ethics of experiments with human subjects
  - Enforced by institutional review systems
- When is it acceptable to deceive subjects?
  - Many security problems naturally include deception

Outline

Email encryption

- Technology became available with PGP in the early 90s
- Classic depressing study: "Why Johnny can't encrypt: a usability evaluation of PGP 5.0" (USENIX Security 1999)
- Still an open "challenge problem"
- Also some other non-UI difficulties: adoption, govt. policy

Phishing

- Attacker sends email appearing to come from an institution you trust
- Links to web site where you type your password, etc.
- *Spear phishing*: individually targeted, can be much more effective

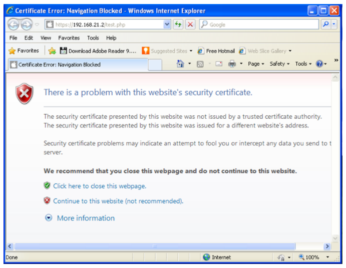
Phishing defenses

- Educate users to pay attention to X:
  - Spelling $\rightarrow$ copy from real emails
  - URL $\rightarrow$ homograph attacks
  - SSL "lock" icon $\rightarrow$ fake lock icon, or SSL-hosted attack
- Extended validation (green bar) certificates
- Phishing URL blacklists

SSL warnings: prevalence

- Browsers will warn on SSL certificate problems
- In the wild, most are false positives
  - `foo.com` vs. `www.foo.com`
  - Recently expired
  - Technical problems with validation
  - Self-signed certificates (HA2)
- Classic warning-fatigue danger

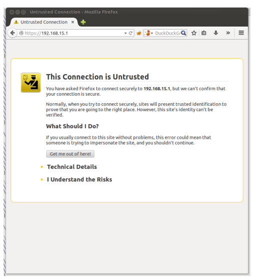
Older SSL warning



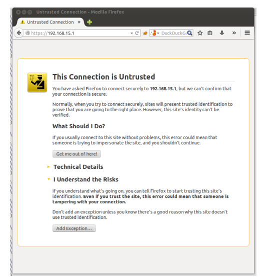
SSL warnings: effectiveness

- Early warnings fared very poorly in lab settings
- Recent browsers have a new generation of designs:
  - Harder to click through mindlessly
  - Persistent storage of exceptions
- Recent telemetry study: they work pretty well

Modern Firefox warning



Modern Firefox warning (2)



Modern Firefox warning (3)

Spam-advertised purchases

- "Replica" Rolex watches, herbal `V!@gr@`, etc.
- This business is clearly unscrupulous; if I pay, will I get anything at all?
- Empirical answer: yes, almost always
  - Not a scam, a black market
  - Importance of credit-card bank relationships

Advance fee fraud

- "Why do Nigerian Scammers say they are from Nigeria?" (Herley, WEIS 2012)
- Short answer: false positives
  - Sending spam is cheap
  - But, luring victims is expensive
  - Scammer wants to minimize victims who respond but ultimately don't pay

Trusted UI

- Tricky to ask users to make trust decisions based on UI appearance
  - Lock icon in browser, etc.
- Attacking code can draw lookalike indicators
  - Lock favicon
  - Picture-in-picture attack

Smartphone app permissions

- Smartphone OSes have more fine-grained per-application permissions
  - Access to GPS, microphone
  - Access to address book
  - Make calls
- Phone also has more tempting targets
- Users install more apps from small providers

Permissions manifest

- Android approach: present listed of requested permissions at install time
- Can be hard question to answer hypothetically
  - Users may have hard time understanding implications
- User choices seem to put low value on privacy

Time-of-use checks

- iOS approach: for narrower set of permissions, ask on each use
- Proper context makes decisions clearer
- But, have to avoid asking about common things
- iOS app store is also more closely curated

Trusted UI for privileged actions

- Trusted UI works better when asking permission (e.g., Oakland'12)
- Say, "take picture" button in phone app
  - Requested by app
  - Drawn and interpreted by OS
  - OS well positioned to be sure click is real
- Little value to attacker in drawing fake button