

CSci 5271
Introduction to Computer Security
Day 22: Anonymizing the network

Stephen McCamant
University of Minnesota, Computer Science & Engineering

Outline

Malware and the network
Denial of service and the network
HW2 walk-through, announcements
Anonymous communications techniques
Tor basics
Tor experiences and challenges

Malware/anti-virus arms race

- "Anti-virus" (AV) systems are really general anti-malware
- Clear need, but hard to do well
- No clear distinction between benign and malicious
- Endless possibilities for deception

Signature-based AV

- Similar idea to signature-based IDS
- Would work well if malware were static
- In reality:
 - Large, changing database
 - Frequent updated from analysts
 - Not just software, a subscription
 - Malware stays enough ahead to survive

Emulation and AV

- Simple idea: run sample, see if it does something evil
- Obvious limitation: how long do you wait?
- Simple version can be applied online
- More sophisticated emulators/VMs used in backend analysis

Polymorphism

- Attacker makes many variants of starting malware
- Different code sequences, same behavior
- One estimate: 30 million samples observed in 2012
- But could create more if needed

Packing

- ▣ Sounds like compression, but real goal is obfuscation
- ▣ Static code creates real code on the fly
- ▣ Or, obfuscated bytecode interpreter
- ▣ Outsourced to independent "protection" tools

Fake anti-virus

- ▣ Major monetization strategy recently
- ▣ Your system is infected, pay \$19.95 for cleanup tool
- ▣ For user, not fundamentally distinguishable from real AV

Outline

Malware and the network

Denial of service and the network

HW2 walk-through, announcements

Anonymous communications techniques

Tor basics

Tor experiences and challenges

DoS versus other vulnerabilities

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

When is it DoS?

- ▣ Very common for users to affect others' performance
- ▣ Focus is on unexpected and unintended effects
- ▣ Unexpected channel or magnitude

Algorithmic complexity attacks

- ▣ Can an adversary make your algorithm have worst-case behavior?
- ▣ $O(n^2)$ quicksort
- ▣ Hash table with all entries in one bucket
- ▣ Exponential backtracking in regex matching

XML entity expansion

- XML entities (HTML `<t`) are like C macros

```
#define B (A+A+A+A+A)
#define C (B+B+B+B+B)
#define D (C+C+C+C+C)
#define E (D+D+D+D+D)
#define F (E+E+E+E+E)
```

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

Malware and the network

Denial of service and the network

HW2 walk-through, announcements

Anonymous communications techniques

Tor basics

Tor experiences and challenges

Virtual network setup

- Every group has unique number GG
- Victim server at
`http://192.168.GG.1/`
- Promptable victim client at
`http://192.168.GG.2/`
- Attacker VM (similar to HW1) at
`192.168.GG.3`
- Use SSH as SOCKS proxy

Q1: basic password

- Plaintext password used for HTTP authentication
- Learn how to use `tcpdump` to read packet contents

Q2: challenge-response password

- HTTP "Digest" authentication based on MD5
- Vulnerable to offline dictionary attack
- Build cracking script to try MD5 construction with different words

HTTPS server

- TLS-enabled server at `https://192.168.GG.1/`
 - Completely different content
- Non-exploited weakness: self-signed certificate
 - Need to click through, etc., in client programs

Q3: predictable cookies

- Cookie used for authentication has no secret or random component
- Reverse-engineer cookie format
- Create fake cookie to impersonate administrative user Stephen
- Figure out how to browse with modified cookie

Q4: SQL injection

- Get access to rows in a database you shouldn't see
- Guess what the query looks like, how to modify it
- Make POST requests not allowed by a form

Q5: stored XSS

- Comments field does not sanitize HTML
- Inject JavaScript in order to steal a cookie
- Victim makes web request to a port you control

HW2 timeline

- VM assignments for registered groups went out in the last couple of hours
- Original due date: 11/26, Tuesday before Thanksgiving
 - Still recommended, now with 10 points extra credit
- Extended due date: 12/1, Sunday after Thanksgiving

Exercise set 4, Q. 1(a) clarification

- Seed random number based on "time of day", `time(2)`
- This includes the date as well: it's the total number of seconds since the beginning of 1970
 - (Cause of potential "year 2038" bug on 32-bit machines)

Exercise sets 2, 3 graded

- Grades for exercise sets 2 and 3 will be on the Moodle momentarily
- Graded papers will be up here after class

Outline

Malware and the network
Denial of service and the network
HW2 walk-through, announcements
Anonymous communications techniques
Tor basics
Tor experiences and challenges

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



Photo credits: Nik Hopper - UPM/CS; Bryan Ford - personal/home page; Ian Goldberg - MIT/Harvard CC-BY-SA 3.0; David Chaum - Marc_Smith on Flickr CC-BY 2.0

Dining cryptographers

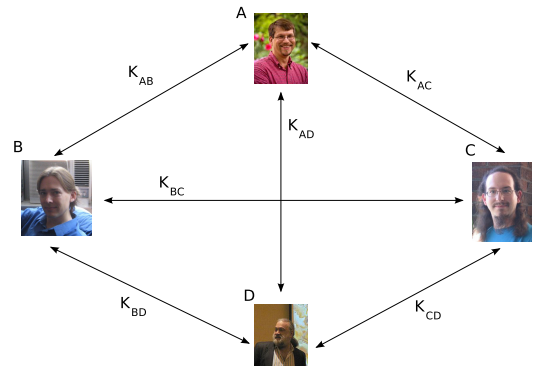


Photo credits: Nik Hopper - UPM/CS; Bryan Ford - personal/home page; Ian Goldberg - MIT/Harvard CC-BY-SA 3.0; David Chaum - Marc_Smith on Flickr CC-BY 2.0

Dining cryptographers

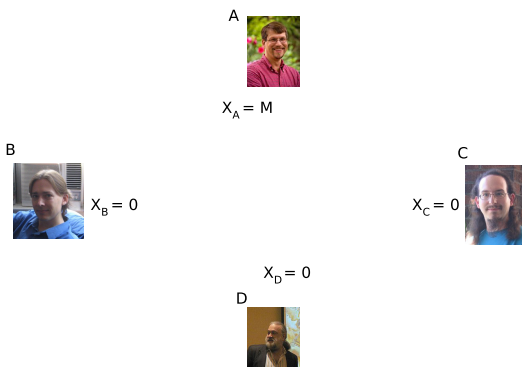


Photo credits: Nik Hopper - UPM/CS; Bryan Ford - personal/home page; Ian Goldberg - MIT/Harvard CC-BY-SA 3.0; David Chaum - Marc_Smith on Flickr CC-BY 2.0

Dining cryptographers

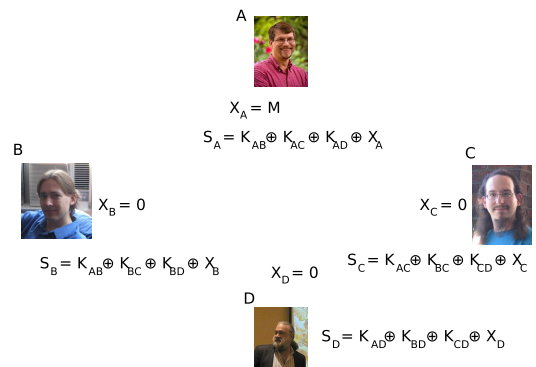





Photo credits: Nik Hopper - UPM/CS; Bryan Ford - personal/home page; Ian Goldberg - MIT/Harvard CC-BY-SA 3.0; David Chaum - Marc_Smith on Flickr CC-BY 2.0

Dining cryptographers

$$\begin{aligned} R &= S_A \oplus S_B \oplus S_C \oplus S_D \\ R &= X_A \oplus X_B \oplus X_C \oplus X_D \\ R &= M \end{aligned}$$

A  $X_A = M$
 $S_A = K_{AB} \oplus K_{AC} \oplus K_{AD} \oplus X_A$

B  $X_B = 0$
 $S_B = K_{AB} \oplus K_{BC} \oplus K_{BD} \oplus X_B$

C  $X_C = 0$
 $S_C = K_{AC} \oplus K_{BC} \oplus K_{CD} \oplus X_C$


D  $X_D = 0$
 $S_D = K_{AD} \oplus K_{BD} \oplus K_{CD} \oplus X_D$

Photo credits: Neil Hopper - UMN CS; Bryan Ford - personal home page; Ian Goldberg - MIT; Henry CC BY-SA 3.0; David Pratum - Marc Smit; Ian Fiske CC BY 3.0

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

Malware and the network
Denial of service and the network
HW2 walk-through, announcements
Anonymous communications techniques
Tor basics
Tor experiences and challenges

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 policies, public keys, etc.
- Replicated, but potential bottleneck for scalability and blocking

Outline

Malware and the network
Denial of service and the network
HW2 walk-through, announcements
Anonymous communications techniques
Tor basics
Tor experiences and challenges

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
- Currently 1-2 sec for 50KB, 5-10 sec for 1MB
- Historically worse for many periods
 - Flooding (guessed botnet) earlier this fall

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 (currently behind)

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" in the news recently

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 → 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 by JS vulnerability patched in June

Next time

- 📌 How usability affects security