L1: Intro, Threat Models

Wednesday, February 8, 2017 1:04 PM

Structure:

- Read papers before class
- Answer reading question
- Ask own question

Labs: 5 labs

last lab: lab 5 or open final project — security analysis

Security

- Goal w/ adversary

Policy - rules/plan

Threat model: Assumptions about what adversary can/cannot do

Mechanism: Enforces policy

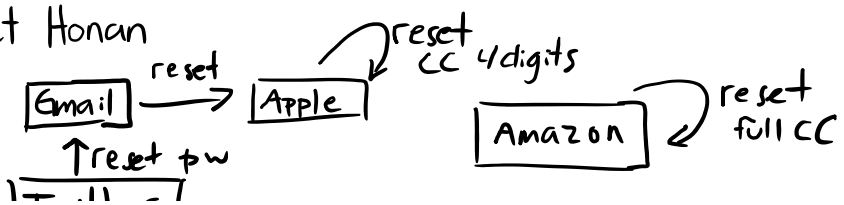
Security = negative goal

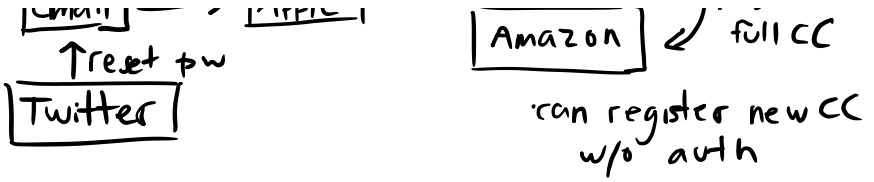
↳ need to take everything into account

Examples

Policy problems:

- Airplane ticket - change after boarding plane
- OCR email images for domain authentication
- Clashing features: teachers can change student's passwords + walk-in registration
- Met Honan





lesson learned: policies can interact in complex ways

Threat models policy

- CPU Power. Kerberos DES 56-bit keys
- CAPCHA - capcha farms in foreign countries $0.001/capcha

Mechanism -Incorrect implementations

- iCloud on web missing ratelimiting

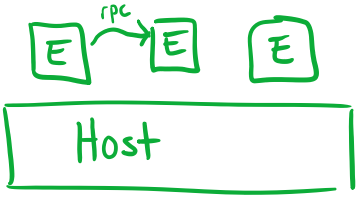
L2: Security Architecture

Wednesday, February 15, 2017 1:06 PM

Google Model

Http → Google Frontend → RPC → ...

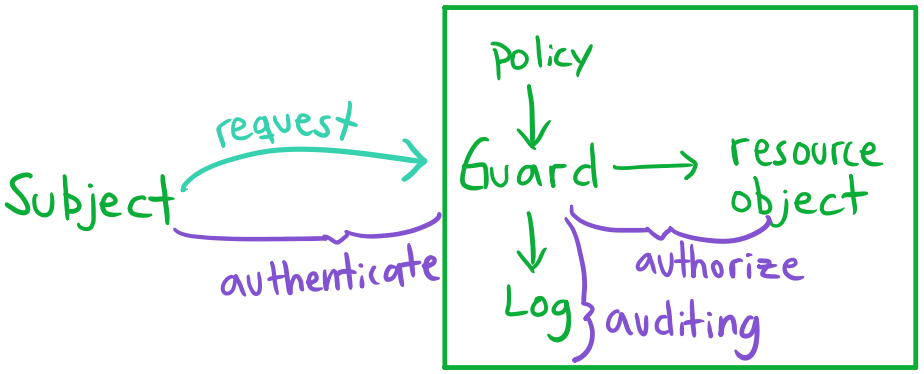
Isolation



Host guarantees environments are isolated from one another

- Virtual machines (HW virtualization)
- Sandbox (kernel sandbox)
- Physical machine → for sensitive services

Guard Model



"The Gold standard"

authenticate
authorize
audit

Authentication

Principal:

- End user person
- Services
- Physical Machines

End user person:

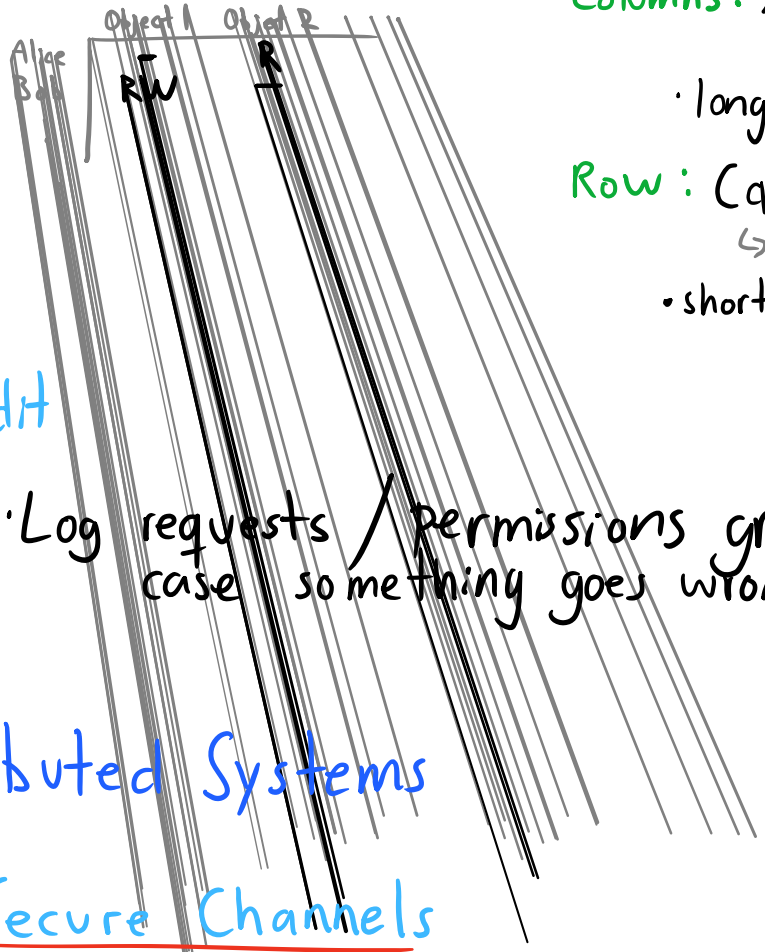
- biometrics
- 2FA
- password
- U2F (not phishable)

Authorize

perms = POLICY(subject, object)

Columns: Access control list (ACL)
- long-term ("AgIs")

Row: Capabilities
↳ great for delegation
- short term policy

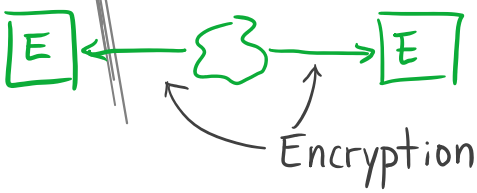


Audit

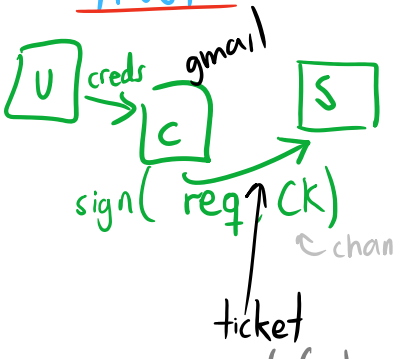
- Log requests / permissions granted in case something goes wrong

Distributed Systems

Secure Channels



Trust



ticket
└(short-term capabilities)

DoS Protection

Goal: Availability

Principles:

- Avoid asymetric resources
  - request should take "as much work" as response
- Authenticate ASAP
  - to establish an online identity
  - Limit/prioratize based on principal
- Split off pre-authentication systems

R1 Buffer overflows

Wednesday, February 15, 2017 8:09 PM

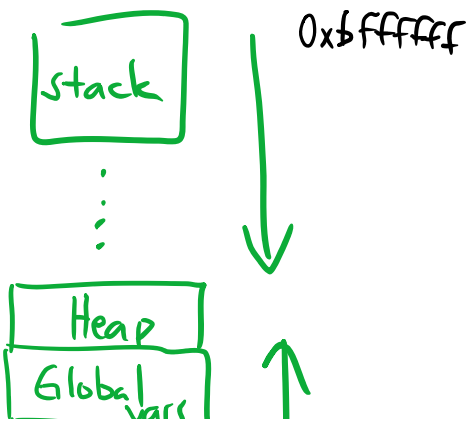
Lab

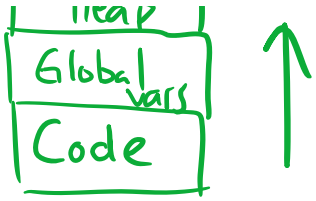
x86 architecture

Registers

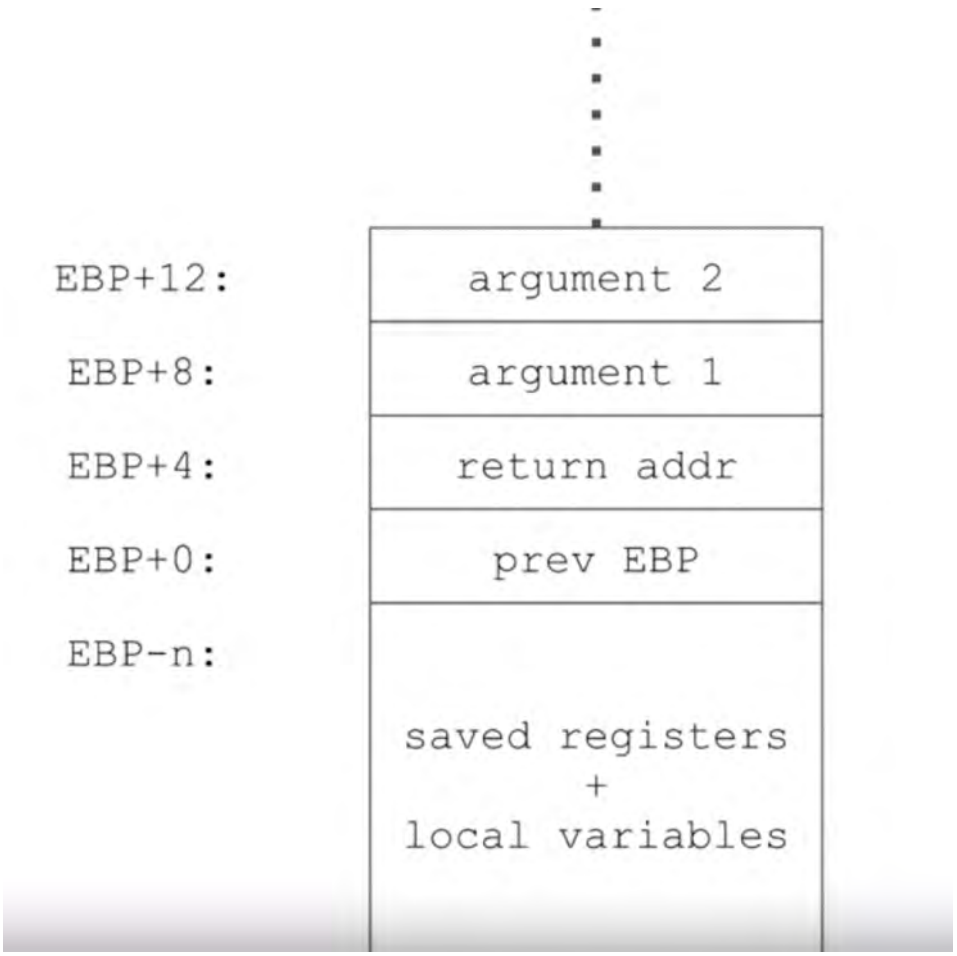
EAX
ECX
EDX
EBX
ESP ← stack pointer
EBP ← Base pointe
  └ Base of current stack frame
ESI
EDI
EIP ← address to be run next

Stack





Calling convention - cdecl



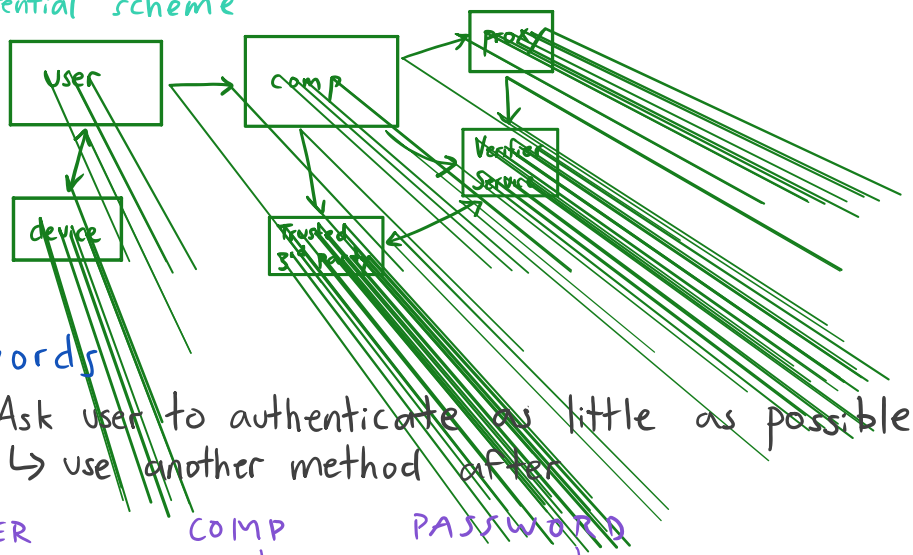
L3: User Authentication

Tuesday, February 21, 2017 1:07 PM

User Authentication

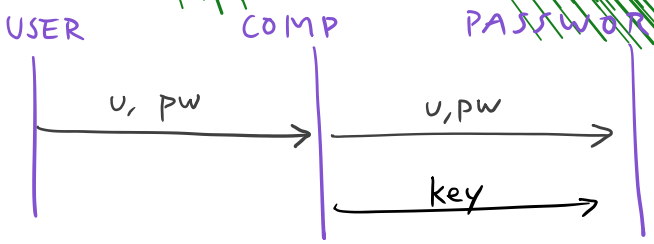
Goal: Verify user's identity

Potential scheme



Passwords

- Ask user to authenticate as little as possible
  - ↳ use another method after



Problems

- Easily guessable
  - 5,000 unique passwords account for 20% of users (32M)
  - ↳ need to rate limit guesses
    - per user AND per adversary
- Format vs entropy ("Password1")
  - Enforcing format does not necessarily increase entropy
- Reused across sites

Storing Passwords

- Don't store plaintext hashes → to prevent against rainbow attacks
  - ↳ store Hash of salted passwords instead
    - ↳ slow hash function

→ store hash of salted passwords instead

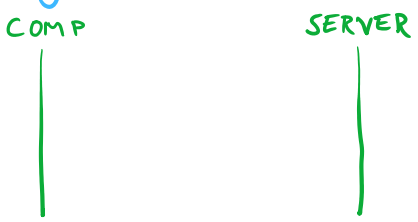
↳ slow hash function

· bcrypt (on purpose)

USER	SALT	Pass Hash
alice	x802	H(salt + pw)
⋮	⋮	⋮

- now a rainbow table can only be used for one user
  ↳ expensive to build

Transmitting Passwords



L4: Control Hijacking attack

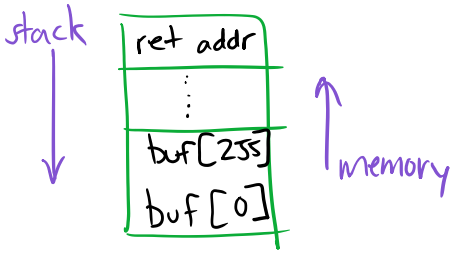
Wednesday, February 22, 2017 1:04 PM

Mitigating Buffer overflows

- Check bounds, be careful
- Avoid C – "Java, Rust, Go, Python"
- Catch bugs / prevent exploits

Buffer Exploit

- Write code onto stack / buffer
- Overwrite ptr (return address)
- Call function pointer (aka return)



Defenses

- NX stack
  - CPU will refuse to execute instructions that live on stack

→ can be circumvented by returning to existing code libc

- Canaries
  - compiler inserts extra code to insert canary into stack and check canary before returning
  - canary can include "\0" terminator + Random stuff stored in global var to defend (since payload with terminator may no longer overflow buffer)

→ There may be other function pointers before the canary.

→ Not all protocols use the same null terminator

- ASLR (address space layout randomization)
  - Randomize memory contents
    - Operating system puts different stuff in different places in memory for each execution.

→ NOP surfing - add a bunch of NOPs to buffer followed by shellcode. IP will keep increasing until shell code reached

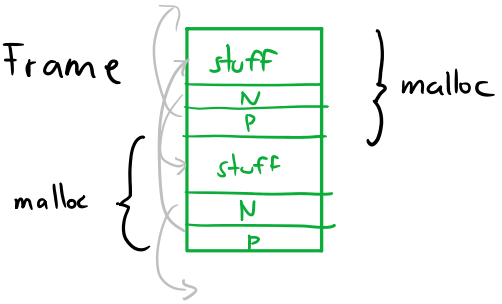
→ Other ways to bypass → even with NX stack

Heap Exploit

Heap Exploit

```
char *p = malloc(16)
gets(p)
```

- Malloc Frame



- doubly linked-list in memory

→ "unthread" frame from doubly linked list

⇒ X → prev = y

attacker sets

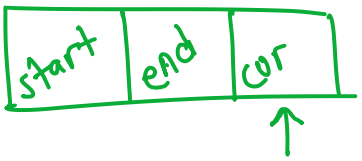
∴ attack goes from there

- Need to understand what is on memory and how it is used

(more sophisticated)

Defenses

Fat Pointer



Alloc: start = cur
end = cur + len

Deref: start ≤ cur < end

Arithm: cur adjust

L5 Priviledge Separation

Monday, February 27, 2017 12:31 AM

L6: Precise Access Control

Wednesday, March 1, 2017 1:07 PM

Need isolated components that support sharing ← gives up isolation?

Confused Deputy (short paper)

- Their compiler gathered statistics and wrote it to /sysx/stat
  ↳ Therfore the compiler can write to sysx

  ⇒ someone can overwrite protected files using the compiler

Principals:

- User : has access to user directory to read code to be compiled
- Compiler: has access to /sysx to write to stats file

- OS allows an operation if it is OK for either principal

How to fix?

- Set up a process for each principal and share messages
- Implement access control checks in the compiler → instead of relying on OS

Why hard?

- Ambient authorities
- Rules complex
- Application-level principal

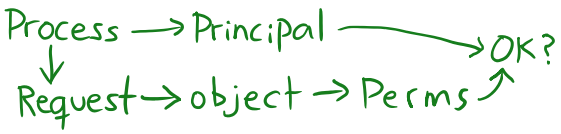
Virtual Machine

+ Strong isolation
+ Unprivileged
+ Unmodified Code

- poor performance
- sharing

Discretionary Access Control (DAC)

- The object owner sets the security policy



+ Users can set their own permissions
+ Fine grained

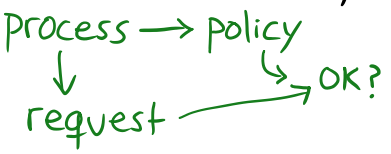
- Permissions spread out
- Net object has no ACLs

+ Users can set their own permissions
+ Fine grained

- Permissions spread out
- Net object has no ACLs
- Need root to create principals
- Need owner to set perms

Mandatory Access Control (MAC)

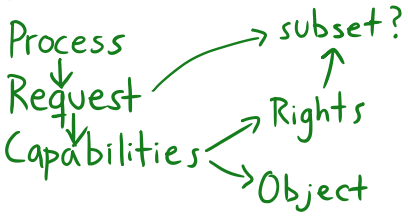
- An admin sets security policy



+ Unprivileged
+ Restrictive
+ Unmodified

- Coarse grained
- missing context

Capabilities

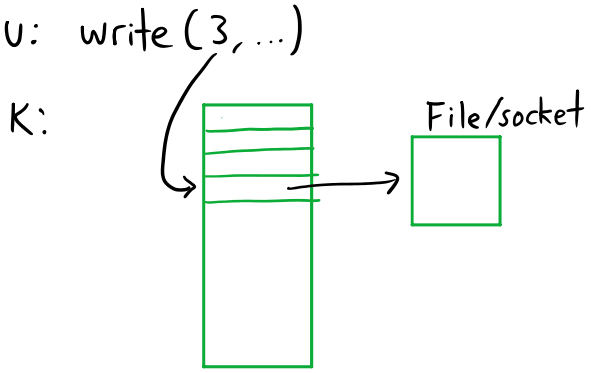


- Every request states the rights it needs for an operation
- No permissions / ACLs
- Unforgeable
- Must be able to delegate a capability

+ Good for short term access control

Unix File descriptors

- Kernel maintains list of File descriptors → files for each process



+ Unforgeable
+ Can be delegated
- No permissions/ACLs

Capsicum

- Application sets up capabilities beforehand
  → calls can enter

Capsicum

- Application sets up capabilities beforehand
  - → calls cap_enter
  - → access via capabilities

+ processes can control their own permissions

How to use it? tcpdump → tries to decode packets on the network → must run as root

① Figure out what capabilities are needed
   ↳ call cap_enter and see what breaks

② cap_enter()

Where Security Bugs come from

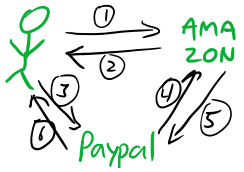
Monday, March 6, 2017 1:06 PM

Guest Lecture: Airbnb security director

Security Model

- What do you care about
  - Data, Money, ... etc
- Probably depends on things you can't control (Gmail → Apple → Amazon Hack) (TSA Checkpoints example)

The Confused Deputy: How to trick an authority into doing something



- CRIME Example
  - Stuxnet

iOS example

- 0-day in iOS ≈ $500K
- 3 0-day exploits ⇒ remote jailbreak + install spyware

- Chrome extension w/ 10,000 users = $10,000 (to install adware w/ update)

L7: Native Client / Software Fault Isolation

Wednesday, March 8, 2017 1:06 PM

Isolation

enforced by:

- Separate machines <=> Physics
- Virtual Machines <=> VMM (Virtual Machine Monitor) (i.e., KVM)
- Processes <=> OS Kernel
- Software ~~Fault Isolation~~ (SFI) <=> Verifier + runtime environment

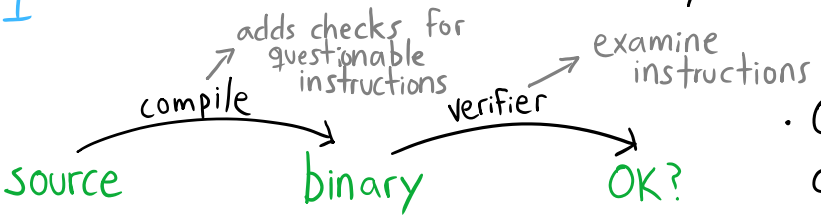
Native Client

Alternatives:

- VM's : Heavyweight
- Capsicum : Buggy OS, platform dependent, incomplete
  - ↳ OS level isolation => used as external sandboxes => security in layers

SFI

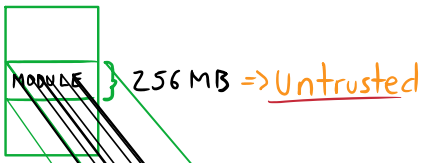
NO: syscall, change seg, access memory



- Questionable instructions checked at runtime using checks added by compiler

Safety

- No syscall instructions
- Memory Safety



Disassembly

- Tricky because of variable length instructions

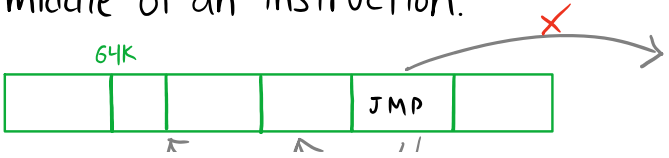
CD 80 = INT $0x80 NOT ALLOWED

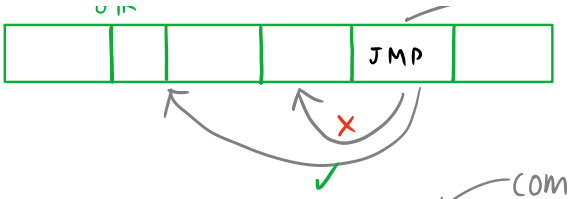
25 CD 80 00 00 = AND %eax $0x0000 80CD ALLOWED

but where does it start for sure?

Reliable dissesembly

- Start at beginning and dissasemble one instruction at a time
- Dissallow jumps ahead of analized and jumps backwards to the middle of an instruction.





compiler can accomplish this by adding NOPs

- Indirect jumps → 32 byte aligned

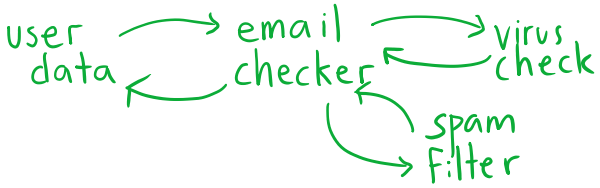
Verifier: Dissasemble every 32 bit multiple

L8: SGX | Ryoan

Wednesday, March 15, 2017 1:04 PM

New

Isolation Goals: Isolating data from untrusted application



Applications use other services ⇒ need to trust all

- can run on unmodified code (IF it uses libc)

Threat Model

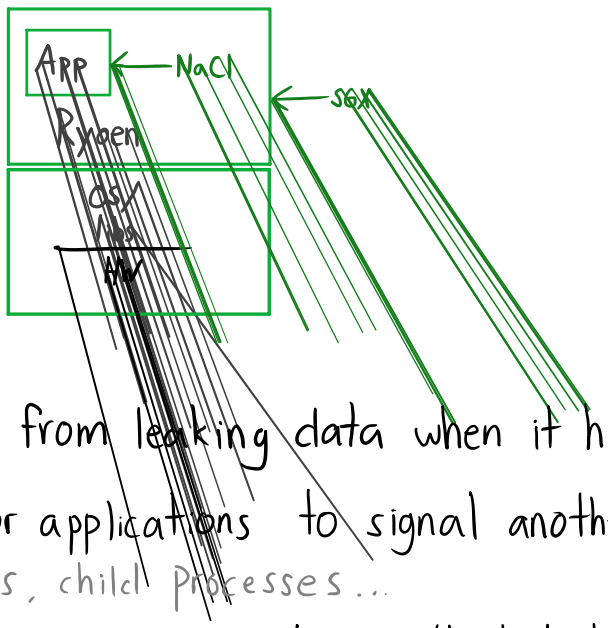
Untrusted Platform

Guard against OS/libraries from interfering w/ execution
- Intel SGX
- Iago attacks

Untrusted app

(has the user's secret data)

Prevent application from leaking user data.
- Native Client for sandbox
- Static I/O - knows what will happen at what time



Leaking data

How to prevent app from leaking data when it has full access.

- Many different ways for applications to signal another process
  - Temp File, I/O timings, child processes ...

⇒ Solution: don't allow application to directly talk to OS after it gets access to user data.

⇒ have to load ANY data it may need during initialization

SGX

- OS Untrusted
- Intel chip trusted, DRAM
- Authenticate/verify everything leaving/entering the chip

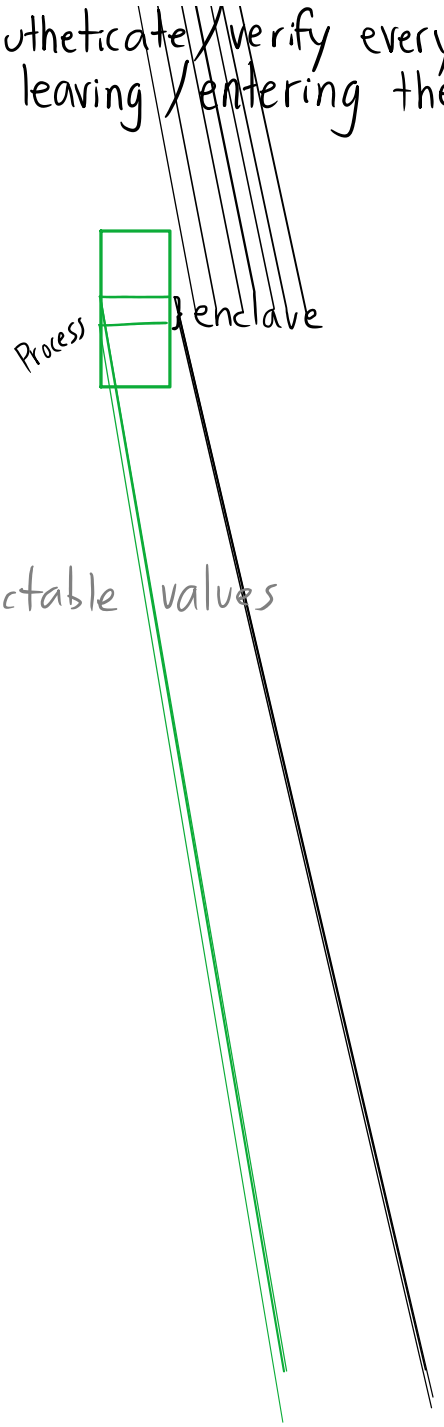
- OS Untrusted
- Intel chip trusted
- Other hardware untrusted (DRAM)

Enclave

- Region of virtual addresses

```
enclave = {
    id: 1
    VAStart: 0x...
    VaEnd: 0x...   }
```

- Authenticate/verify everything leaving/entering the chip



Iago Attacks

- /dev/urandom - OS could return predictable values
- mmap → malloc

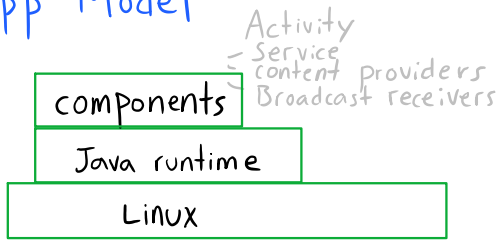
L9 Android Security

Monday, March 20, 2017 1:08 PM

Android

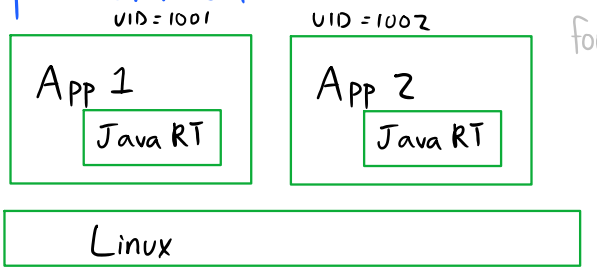
- Users install apps → could be buggy or malicious

App Model



App Isolation

Use UID for isolation



Shared devices

GPS, Camera → dev/gps, dev/camera → files with correct permissions

- Android gives each device a specific GID (Network, SD card)
- Some permissions map directly to a GID
  - ↳ As defined in the manifest

App storage

- Internal Storage /flash all accesses checked by Android => TRUSTED
- SD Card - Any app can access.
  - Modify code / copy paid apps => Per-app key on internal Flash

Intents - defined in manifest

- Messages
  - component: package name of receiver (optional - let user pick based on other fields) ↳ not authenticated
  - action: i.e DIAL, MAIN
  - data: URI ie tel: 123456789
  - category

Permissions

Label: string → what you need to call component

- Every component gets a label
- Each application has a list of labels => privilages the app has

Defining Permissions

Name (DIAL_PERM) → not authenticated => "first come first serve"

Defining Permissions

- Name (DIALPERM) ← not authenticated ⇒ "first come first serve"
  - user-visible name
  - description
  - type: {normal | dangerous | signature}
    - doesn't appear to user.

Takeaways

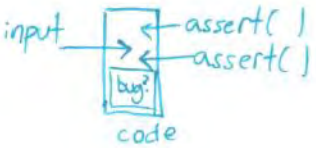
- Android is a huge step forward for security → much better than Windows
- All apps have access to Linux kernel ⇒ could exploit a bug in Linux

L10 Bug finding

Wednesday, April 5, 2017 10:54 AM

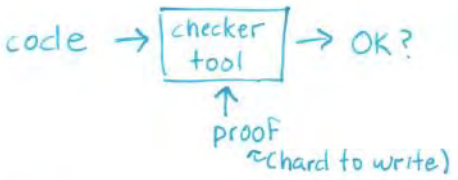
Bug-Finding

- Find bugs before they cause damage



  - Need to know what an error is
    - ⇒ add assert( )'s in call
      - ↳ some asserts (ie divide by 0) can be added automatically

Prove-correctness



Spec ≈ no asserts fail

Testing

inputs, assert(outputs) ← Written manually

+ expected bugs
+ regression
  - ↳ add new checks for new bugs

- unexpected behaviors
- incomplete
- coverage

Fuzzing

- Try random inputs
- write code to generate inputs ⇒ domain-specific fuzzers

+ could find unexpected bugs
  - ? coverage

- still needs asserts

Symbolic Execution

- Makes more educated input guess than fuzzing

EXAMPLE PROGRAM

```
(1)  read x, y
(2)  if x > y :
(3)      x = y
(4)  if x < y
(5)      x = x + 1
(6)  if x + y == 7      } assert(x+y≠7)
(7)      error( )
```

- Run program in a different environment
  - ⇒ memory location = symbolic expression (instead of concrete bits of result)
- Path condition
  - describes if statements followed in execution

(6) if x+y == 7 > assert(x+y ≠ 7)
(7) error()

followed in execution
(widely used in industry)

of result)

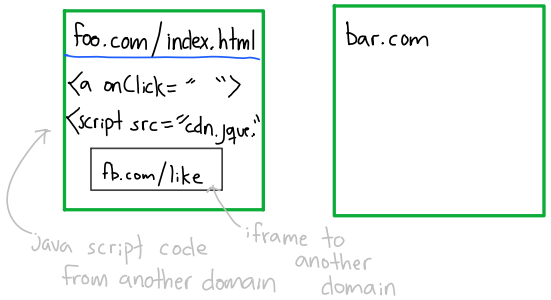
1/

L11 Web Security

Wednesday, April 5, 2017 1:08 PM

Threat Model

- Attacker has own domain
- Attacker running page in user's browsers ← links, ads
- Network is tamperproof
- No implementation bugs in browser



DOM tree Document Object Model

Principal: origin = <protocol, host, port>

http://mit.edu :80

↳ like unix UID.

Same Origin Policy

contain and / run code

Actor (process): window/iframe : origin based on URL

Objects (resources): origin

Script in frame runs w/ frame's origin

Access object if same origin

Resources

DOM nodes

Window

Devices (webcam, mic)

Cookies

Local Storage

Network

Cookies

↳ Application defined blob of data

↳ Application defined blob of data

Browser

DOMAIN	PATH	DATA
*.mit.edu	/6858	~

- Browser stores table of cookies
  Browser sends any matching cookies in request

HTTP Request / Response

- can allways issue requests anywhere
  ⇒ CSRF (cross-site request forgery)

L12: Network Security

Wednesday, April 12, 2017 1:09 PM

Desirable Properties:

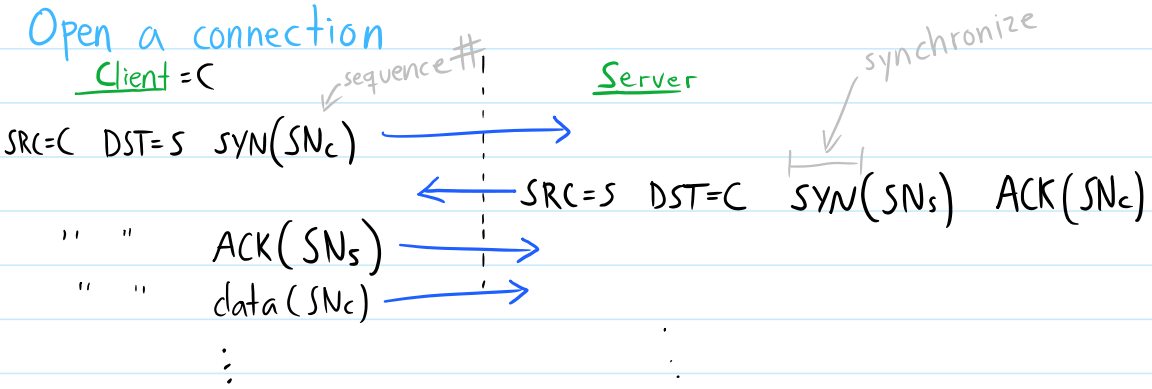
- Confidential
- Availability
- Authentication
- Privacy

Remote login protocol

- Supply credentials ⇒ get sent in plain text
- or Supply list of allowed IPs ⇒ can log in without password from IP

TCP

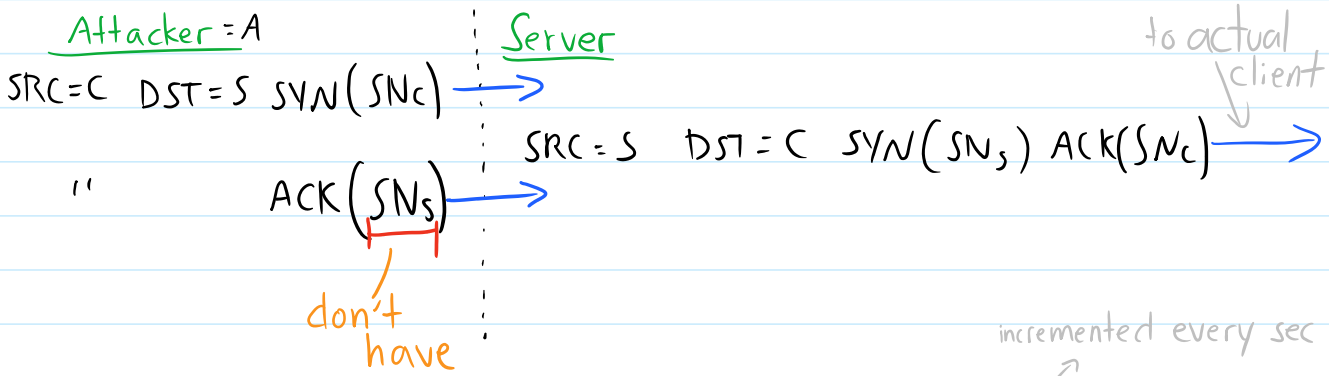
Open a connection



Problems

- Machine constructs entire packet
  ⇒ can set SRC = to anything

Attack:



Guess sequence #: generated by server deterministically (incremented every sec)
⇒ easy to guess

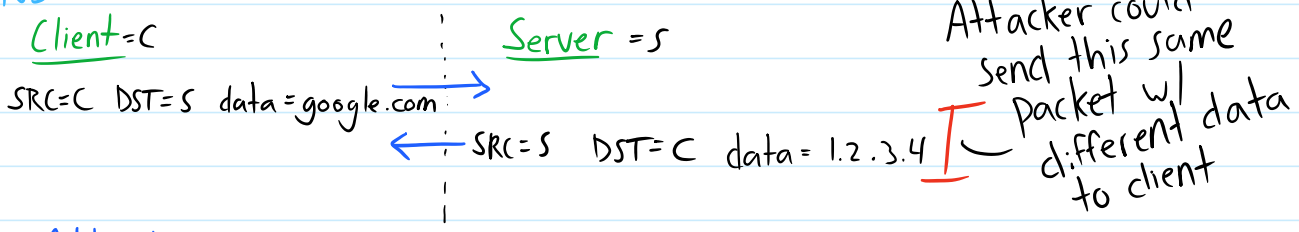
Problems

Problems

- IP based authentication doesn't work
- can inject data if you can guess client sequence #
- DoS: TCP reset

UDP – connection less ⇒ no sequence

DNS



DoS Attacks

google.com — all IP's

Can use DNS: small query and large response ⇒ Amplified bandwidth

- send many DNS requests with SRC = C
  ⇒ DNS Server will send many requests to client

Defenses

symmetry: don't allow amplification by requiring requests that are proportional in length to response

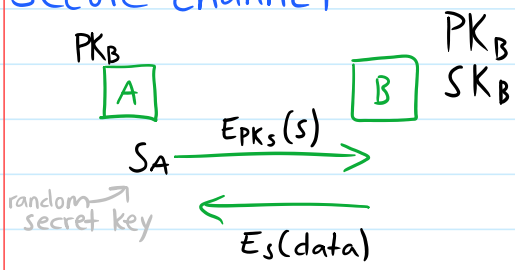
TLS HTTPS

Wednesday, April 26, 2017 1:10 PM

Crypto primitives

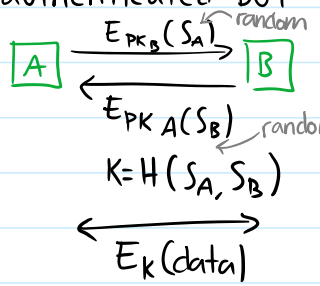
Symmetric and Public Key
fast ↳ slow

Secure channel



- Use PK to get symmetric key to use for further messages
- B is authenticated but A is not

Fix



prevents replay attack since both parties choose new keys for each connection

Forward Secrecy

- At the end of communication no one can reconstruct messages
  - Use temp PK to exchange symmetric key
    → forget/delete everything at the end

Certificates

$Cert = \{ Sign_{SK_{CA}}(B, PK_B), B, PK_B \}$

Revocation

Expiration ~ 1 year

CRL - (certificate revocation list): URL on CA's server → Clunky, often not used

Mixed Content

https sites could include non-https resources

⇒ Attacker could modify http response and add his/her own code

Side-channel attacks

Monday, May 1, 2017 1:05 PM

- Real systems leak information during operations
  - Timing, RF, power, Audio, packet size
- Interested in small secrets of high importance

Security Economics

Wednesday, May 3, 2017 1:09 PM

- Where does the money come from?

Resources
- Compromised system
- Tools
- Stolen data

- End user
  - Buy something
  - defraud

L20: Anonymous Communication

Wednesday, May 10, 2017 1:02 PM

w/ Nick Mathewson nickm@torproject.org

Preliminaries:

- Nick started building Tor in 2003
- ~2M IP's per day: ~1Tb/s
- Why build Tor?
  - Harder to ban technologies that are already out there

IS&T Security

Monday, May 15, 2017 1:15 PM

- In a 24-hour period, MIT gets thousands of intrusion attempts from all around the world
- Greatest weakness to security = humans
  → no matter how good the technology is

Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.858 Fall 2015

Quiz II

You have 120 minutes to answer the questions in this quiz. In order to receive credit you must answer each question as precisely as possible.

Some questions are harder than others, and some questions earn more points than others. You may want to skim them all through first, and attack them in the order that allows you to make the most progress.

If you find a question ambiguous, be sure to write down any assumptions you make. Be neat and legible. If we can't understand your answer, we can't give you credit!

Write your name and submission website email address on this cover sheet.

This is an open book, open notes, open laptop exam.
NO INTERNET ACCESS OR OTHER COMMUNICATION.

Please do not write in the boxes below.

I (xx/12)	II (xx/3)	III (xx/9)	IV (xx/9)	V (xx/13)	VI (xx/9)	VII (xx/12)	VIII (xx/2)	Total (xx/69)

Name: Fernando Trujano Test #1

Submission website email address:

I Web security

Ben Bitdiddle runs a popular web site at `bensblog.com`. Ben decides he wants to make some money on the side, and signs up with a third-party ad network. The ad provider gives him the following HTML snippet ("snippet 0"), and tells him to add it to his page where he wants ads to be shown.

```
<script src="http://adsadsads.com/ads.js?customer=bensblog">
</script>
```

Ben has taken 6.858 and decides to use HTTPS instead of HTTP to fetch the script ("snippet 1"):

```
<script src="https://adsadsads.com/ads.js?customer=bensblog">
</script>
```

Ben's site is also served over HTTPS. Both sites have valid and trusted TLS certificates, and their private keys have not been stolen.

1. [3 points]: Describe one attack that is possible with snippet 0 that is prevented by snippet 1? (Briefly explain your answer.)

An attacker snooping on the network cannot serve a malicious script when someone visits ben's site because of https.

Ben reads about the `integrity` attribute. This is part of a security mechanism called *Subresource Integrity (SRI)* that is currently being standardized. An `integrity` attribute can be added to a `script` tag to indicate the expected cryptographic hash for the script being referenced. If the contents of the script that the browser downloads do not match the given hash, the browser will refuse to execute the script. The intent is that an SRI integrity check should be added to any script loaded from a third-party source to ensure that it has the expected content.

Ben adds an `integrity=` attribute to snippet 1 with the hash of the current script, so that his web pages instead include this snippet ("snippet 2"):

```
<script src="https://adsadsads.com/ads.js?customer=bensblog"
  integrity="sha384-R4/ztc4ZlRqWjqI...ZETq72KgDVJCp2TC">
</script>
```

2. [3 points]: Describe an attack that snippet 1 allows, but snippet 2 prevents. (Briefly explain your answer.)

IF adsadsads.com gets compromised, an attacker could change ads.js and bens site would execute + w/ snippet 1 but not snippet 2 (since the hash changed)

Ben is concerned with the loading time of his site, and opens up his browser's network request inspector to see how he might speed things up. He sees that the TLS handshake with the ad provider is taking several milliseconds. Ben figures that it is sufficient for his site alone to be served over HTTPS, and decides to further change the ad code to use HTTP instead of HTTPS to avoid the overhead of an additional TLS connection. Ben's ad code now looks like this ("snippet 3"):

```
<script src="http://adsadsads.com/ads.js?customer=bensblog"
  integrity="sha384-R4/ztc4ZlRqWjqI...ZETq72KgDVJCp2TC">
</script>
```

3. [3 points]: Is snippet 3 more secure, less secure, or equivalent in security to snippet 2? (Briefly explain your answer.)

It is less secure in that an adversary on the network can see site activity with adsadsads.

Ben modifies his web site to have a login form. After a successful login, the site stores a cookie with the user's credentials in the user's browser. Ben runs this plan by Alyssa and she recommends that he arrange with the ad network to be able to put an iframe around snippet 1 as follows ("snippet 4"):

```
<iframe src="https://adsadsads.com/ads.html?customer=bensblog">
</iframe>
```

`https://adsadsads.com/ads.html?customer=bensblog` contains:

```
<script src="https://adsadsads.com/ads.js?customer=bensblog">
</script>
```

4. [3 points]: Give an example of an attack that is possible without the iframe (snippet 1) but impossible with the iframe (snippet 4).

? ~~CSRF~~? - script in frame can't access cookies if ads.js compromised

II Ur/Web

Ben is excited about Ur/Web, as described in the paper "Ur/Web: A simple model for programming the web" by Chlipala. He rewrites the `zoobar` application from Lab 4 using Ur/Web.

5. [3 points]: Identify a specific security exploit from Lab 4 that would be eliminated by using Ur/Web. Briefly explain why Ur/Web would eliminate that attack.

III SYN Floods

Recall the TCP discussion in Lecture 12: before 1996, server TCP implementations would handle arrival of a SYN (connection setup) packet by remembering the connection, and by replying with a SYN+ACK packet containing the server's initial sequence number (ISN). The server expects an ACK packet from the client echoing the server's ISN. The server would keep state for this "half-open" connection for many minutes. To avoid running out of memory, server TCP implementations would limit the number of half-open connections allowed to exist; if a SYN arrived when too many already existed, the server would ignore the SYN (and thus not allow a connection to be set up).

In the "SYN flood" denial-of-service attack described in the lecture the attacker sends TCP SYN packets to a server at a low rate, perhaps a few per second. These SYN packets have random source IP addresses — that is, the source is forged. The attacker would not send the ACK packets the server expected. The result is that the server almost always had the maximum number of half-open connections, and thus ignored legitimate client connection requests.

A charming (for the attacker) feature of this attack is that it required only a very low rate of attack packets, so it was easy to mount the attack from a single computer with a cheap Internet access link. Further, the low volume would likely not raise any alerts from network monitoring systems looking for large volumes of traffic.

6. [3 points]: Suppose Bob is aiming a SYN flood attack from his home computer against www.anti-bob-club.com. Why can't the server at www.anti-bob-club.com recognize the packets from Bob's computer and ignore them?

Packets don't contain Bob's IP address and could be from anyone.
↳ ie can send packets with any IP as "source"

The "SYN cookie" defense described in Lecture 12 defeats SYN flood attacks. The idea is that the server's TCP responds to a SYN packet with a SYN+ACK packet containing the following ISN (some details omitted):

```
ISN = SHA1(source IP address, secret)
```

The secret is a value that only the server knows; the source IP address is the source address in the SYN packet. The server does *not* keep any state about this half-open connection, and thus does not limit the number of half-open connections that can exist. If an ACK arrives at the server for a connection that doesn't exist, the server checks whether the sequence number being acknowledged is equal to the ISN computed above. If it is, the server concludes that the ACK is part of a legitimate connection setup sequence, and the server creates state for a new connection. Otherwise the server ignores the ACK packet.

www.anti-bob-club.com uses SYN cookies.

7. [3 points]: Bob thinks maybe he can just wait for the server's SYN+ACK packet to arrive, look at the ISN it contains, and send that ISN back to the server in an ACK. That would cause the server to allocate memory for a new connection; if Bob created enough connections like this, and never closed them, perhaps he could force the server to run out of memory. Why won't Bob's idea work?

In order to get the SYN-ACK packet, Bob would have to use a legitimate IP address, which could be blocked.

8. [3 points]: Explain why the secret is an important part of the server's ISN computation.

Without the secret anyone can get an ISN for any IP Address

IV Kerberos

Alice, an MIT student, is designing a secure lab submission system. The goal is to let students in various courses submit their lab solutions for grading. Alice's design gives each course its own lab submission server. It's important that students upload their labs to the correct server for their course, to avoid unwanted disclosure of their solutions. For example, if rogue students were to set up fake submission servers, Alice's system should not accidentally upload solutions to those servers.

Alice wants to use Kerberos to provide security (see *Kerberos: An Authentication Service for Open Network Systems*). She gives each course's server its own Kerberos principal. For example, 6.858's server has Kerberos principal "UP858," the Kerberos KDC has an entry for "UP858" with a corresponding key, and 6.858's server knows that key. Alice writes an application that students can run on Athena workstations that uses Kerberos to establish a secure channel to a submission server. Assume that the cryptography underlying Kerberos secure channels ensures confidentiality and integrity.

Students need a way to tell Alice's application the correct server to which to upload their lab. Alice thinks it would be great if instructors could write the information students need on the blackboard on the first day of class.

9. [3 points]: Can Alice design the system in this way? If yes, what should instructors write on the board, and why does that work? If no, why not?

Yes. write the Kerberos principal UP858. A malicious server will not have the UP858 key

Suppose the MIT AFS server is down, but Bob doesn't know that. Bob's workstation uses Kerberos to try to set up a connection to the AFS server. Eve sets up a fake server that sends and receives packets using the AFS server's IP address; the LAN sends the packets Bob addresses to the AFS server to Eve's server instead. Eve runs her own software on her server, which mimics the real MIT AFS server as best it can, though it does not know the real AFS server's Kerberos key. Eve cannot compromise the KDC.

Please answer the next two questions based on the Kerberos protocol as described in the paper (Section 4) and/or the notes for Lecture 13. (If you are familiar with the real Kerberos protocol implementation, you may realize that some of its details could affect the answers; but please answer with respect to the paper's description.)

10. [3 points]: At what point in the Kerberos protocol (if any) will Bob's workstation realize that there is a problem?

Now suppose that Eve can also modify Bob's packets on the LAN. Eve modifies Bob's message to the TGS server (Figure 8 in the paper, or step 3 in the notes) to mention "eve@mit.edu" instead of the Kerberos ID of the AFS server; note that the server name in this message is not covered by encryption. Eve *does* know the Kerberos password for eve@mit.edu.

11. [3 points]: At what point in the Kerberos protocol (if any) will Bob's workstation realize that there is a problem?

V TLS and HTTPS

12. [7 points]: Assume that a TLS connection has been established successfully between a client and a server. Establishing the session included checking the server certificate and executing a Diffie-Helmann exchange, but the client did not provide a client certificate. Further, assume that the client and server are honest, that the client and server don't leak their keys, and that the cryptography is good. Which of the following attacks does TLS protect against?

(Circle True or False for each choice.)

A. **True / False** An attacker replacing bytes sent by a client with bytes of the attacker's own choosing.

B. **True / False** An attacker reading the plaintext bytes sent by a client.

C. **True / False** An attacker replaying bytes that a client sent earlier.

D. **True / False** An attacker impersonating the server.

E. **True / False** An attacker impersonating the client.
once session starts

F. **True / False** An attacker stealing the server's private key and reading the plaintext of recorded past connections.
forward secrecy

G. **True / False** An attacker breaking into a certificate authority and creating a fake certificate for the server.

Consider a browser that retrieves a web page over HTTPS. A Web developer by accident incorporated a script in that page as follows:

```
<script src="http://jquery.com/jquery1.4.1.min.js"></script>
```

13. [3 points]: What does ForceHTTPS (as described in the paper by Jackson and Barth) do to avoid such accidents? (Explain your answer briefly.)

Will give a mixed-content error

14. [3 points]: Ben modifies FireFox to turn every certificate error into "a page not found" error message. He reasons this improves security because it prevents users from clicking through invalid certificates. Many users download his modified browser, but quickly stop using it. Why would users want to give up on the better security offered by Ben's browser?

some legitimate sites have misconfigured certificates

Users prefer usability over security

VI Side channels

15. [6 points]: The openSSL implementation described in "Remote Timing Attacks are Practical" (by Brumley and Boneh) uses the following performance optimizations: Chinese Remainder (CR), Montgomery Representation (MR), Karatsuba Multiplication (KM), and Repeated squaring and Sliding windows (RS). Which of the following options would close the timing channel attack described in the paper if you turned the listed optimizations **off**?

(Circle True or False for each choice.)

A. True / False CR, MR, KM, and RS.

B. True / False RS

C. True / False RS and KM

D. True / False RS and MR

E. True / False CR and MR

F. True / False CR

16. [3 points]: Ben launches the remote timing attack described in the paper from MIT on a server running at Stanford. He produces a graph in the style of Figure 3a of the paper, but he is unable to guess q. Why is that the case? How might his graph look like? (Briefly explain your answer.)

The RTT is too much to accurately measure timing.

VII Tor

Fred wants to make a complaint about 6.858 to the staff, but he doesn't want the staff to know the complaint is from him. He connects from a PC in his dorm room through Tor to the 6.858 complaint-submission web server and submits some text. By a cruel twist of fate, the first Onion Router (OR) in the multi-OR Tor circuit he sets up is controlled by the 6.858 staff. Thus the 6.858 staff can observe that Fred is creating a Tor circuit, since it comes from Fred's IP address. The 6.858 staff have no special powers with respect to Tor other than this.

17. [3 points]: Can the first OR in Fred's circuit tell where Fred is connecting to? How, or why not?

No, the first OR only knows about the second OR. The rest is encrypted in layers for the next ORs

18. [3 points]: Are the 6.858 staff likely to be able to tell that the complaint submitted to their server is really from Fred, despite his use of Tor? Why or why not?

Probably because the timing of Fred's connection to Tor and the complaint will be similar.

Alice uses Google Mail (gmail) through Tor. Bob claims that her use of Tor is pointless, since Alice has to log into gmail, so she can't be anonymous to gmail regardless of Tor.

19. [3 points]: Explain how Alice might benefit from using Tor despite Bob's objection.

- Changes the IP
- can hide from network snoopers / ISPs

20. [3 points]: Butler Lampson explained there is an opposition between security and convenience, and that convenience in general wins. Can you give an example in the design of Tor where convenience trumped security? (Briefly explain your answer.)

Tor could delay traffic to make it harder to deanonimize connections like in 18) but that would be too inconvenient.

Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.858 Fall 2014

Quiz II

You have 80 minutes to answer the questions in this quiz. In order to receive credit you must answer the question as precisely as possible.

Some questions are harder than others, and some questions earn more points than others. You may want to skim them all through first, and attack them in the order that allows you to make the most progress.

If you find a question ambiguous, be sure to write down any assumptions you make. Be neat and legible. If we can't understand your answer, we can't give you credit!

Write your name and submission website email address on this cover sheet.

This is an open book, open notes, open laptop exam.
NO INTERNET ACCESS OR OTHER COMMUNICATION.

This quiz is printed double-sided.

Please do not write in the boxes below.

I (xx/16)	II (xx/24)	III (xx/10)	IV (xx/16)	V (xx/8)	VI (xx/12)	VII (xx/8)	VIII (xx/6)	Total (xx/100)

Name: Fernando Trujano Test #2

Submission website email address:

I User authentication

1. [8 points]: A company named Vault wants to offer a secure, cloud-based backup system. When the user updates a local file, her Vault client opens a TCP connection to a Vault server, and uses the Diffie-Helman protocol to establish a secret symmetric key K with the server. Then, the client generates this string s:

$$s = \langle \text{ documentName, documentContent, userName, userPassword, randomNumber } \rangle$$

and sends the following message to the Vault server:

$$E_K(s, \text{HMAC}_K(s))$$

where $E_K(m)$ denotes encrypting message m using key K, and $\text{HMAC}_K(m)$ denotes computing an HMAC message authentication code of message m using key K.

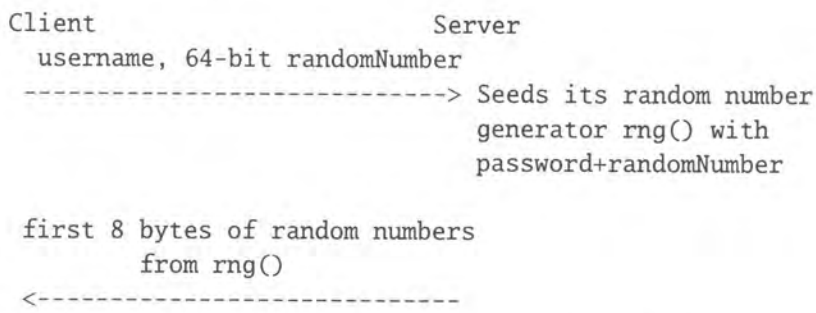
The server decrypts the message, verifies the user's password, and verifies the integrity of the message using the HMAC. If all of the checks succeed, the server stores the document. If the server sees more than 10 messages with the wrong password, all future accesses to that account are blocked.

How can a network attacker reliably obtain the user's password?

Man in the middle: impersonate Vault server to steal password

2. [8 points]:

Suppose that a user wants to verify that a server knows the user's password. The user and the server engage in the following challenge/response:



Everyone knows which algorithm the server uses for `rng()`, so the client can validate that the server's response contains the expected bytes.

Suppose that many different servers use this protocol. Further suppose that the attacker has a list of popular usernames, and a separate list of popular passwords. Explain how an attacker can abuse the protocol (without otherwise compromising any servers) to build a rainbow table of passwords and easily determine the passwords of many users.

map 8bytes random numbers → password

Always send the same randomNumber to Server

II Tor and Private browsing

Imagine that the website `https://foo.com` embeds a JavaScript file from `http://attacker.com`. Suppose that a user's browser allows an HTTPS page to run JavaScript code fetched from an HTTP URL; however, the code cannot read or write any persistent client-side state. For example, the JavaScript code cannot read or write cookies, nor can it read or write DOM storage. The browser also ensures that the `attacker.com` web server cannot set client-side cookies using the `Set-Cookie` header, or receive client-side cookies in the HTTP request for the JavaScript file.

Suppose that the user visits `https://foo.com` once in private browsing mode, closes the tab, and then visits the site again in regular browsing mode; in both cases, the user's web traffic goes through Tor. The `attacker.com` web server would like to determine with high likelihood that the same user has visited the site twice. However, the attacker does not control any Tor nodes.

3. [8 points]: Why is it unlikely that the `attacker.com` server can use TCP fingerprinting to identify the user? Recall that TCP fingerprinting involves looking at TCP connection parameters, such as the initial TCP window size, TCP options, TCP flags, etc.

The Tor exit node and the client can have different TCP fingerprints

4. [8 points]: Describe a way that the attacker can fingerprint the user with a much higher likelihood of success.

- Force client to load an image, if both connections load it ✓

- Send JS info to attacker
  ↳ Resolution, name, etc

5. [8 points]: Browsing the web through Tor can be slow. This is because user traffic is forwarded between volunteer computers that are scattered across the world, overburdened with traffic, and potentially situated behind slow network connections.

Suppose that CloudCo, a large technology company with datacenters scattered across the world, offers some of its machines to the Tor community for use as entry and exit nodes. CloudCo machines have plentiful RAM and CPU; CloudCo machines also have low latency, high-bandwidth network connections to all major ISPs. By using CloudCo machines as Tor entry and exit nodes, users could ensure that Tor congestion would only exist in the middle of a circuit.

Assume that CloudCo genuinely wants to help Tor users, and that CloudCo configures its machines to faithfully execute the Tor protocol. Why is it still a bad idea for users to employ CloudCo machines as entry and exit nodes?

Loses anonimity by looking at Timings

III TaintDroid

6. [10 points]: TaintDroid defines various sources of taint, and a unique taint flag for each of those sources (e.g., `IMEI`, `PHONE_NUM`, `CAMERA`, etc.).

For the application code below, list the set of taint flags that each variable will have *after* each line of code has executed (for example, `"IMEI, CAMERA"` or `"PHONE_NUM"`). If a variable will not have any taint flags, write an Ø.

```
int x = android.getIMEI();                    x:_____
int y = android.getPhoneNum();                y:_____
int z;
if(x > 5000){
    z = 0;                                    z:_____
}else{
    z = 1;                                    z:_____
}

int arr[] = new int[2];                       arr:_____
arr[0] = x;                                   arr:_____
arr[1] = y;                                   arr:_____

char buf[] = android.getCameraPicture();      buf:_____
int val = buf[x%2];                           val:_____

x = 42;                                       x:_____
```

IV Timing side channels

Consider the timing attack on OpenSSL RSA keys, described in the paper by Brumley and Boneh.

7. [8 points]: Suppose that OpenSSL was modified to never use Karatsuba multiplication (and instead always use "normal" multiplication), but was still using Montgomery multiplication as described in the paper. Would you expect the attack to still work with a few million queries? Explain why or why not.

8. [8 points]: Suppose that OpenSSL was modified to never use Montgomery multiplication, but was still using both Karatsuba and normal multiplication as described in the paper. Would you expect the attack to still work with a few million queries? Explain why or why not.

V Embedded device security

9. [8 points]: Ben Bitdiddle has a smartphone with an always-on voice recognition system, which runs any commands that it hears.

Alyssa P. Hacker wants to trick Ben's phone into running a command, but Ben turns off all wireless radios on the phone as a precaution to prevent Alyssa from breaking in over the network, and also keeps his phone in a locked sound-proof room in hopes of foiling Alyssa. After hearing Kevin Fu's guest lecture, Alyssa figures out how she can get Ben's phone to run a command of her choice, without breaking into Ben's room. What is Alyssa's plan?

Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.858 Fall 2013

Quiz II

You have 80 minutes to answer the questions in this quiz. In order to receive credit you must answer the question as precisely as possible.

Some questions are harder than others, and some questions earn more points than others. You may want to skim them all through first, and attack them in the order that allows you to make the most progress.

If you find a question ambiguous, be sure to write down any assumptions you make. Be neat and legible. If we can't understand your answer, we can't give you credit!

Write your name and submission website email address on this cover sheet.

This is an open book, open notes, open laptop exam.
NO INTERNET ACCESS OR OTHER COMMUNICATION.

This quiz is printed double-sided.

Please do not write in the boxes below.

I (xx/14)	II (xx/8)	III (xx/9)	IV (xx/14)	V (xx/6)	VI (xx/18)	VII (xx/18)	VIII (xx/9)	IX (xx/4)	Total (xx/100)

Name:

Submission website email address:

This page intentionally left blank.

I Web security

1. [8 points]:

Recall that Zoobar has a cross-site scripting vulnerability that allows an adversary to construct a URL so that the server's response contains arbitrary Javascript code (from the URL itself). In lab 5, you used this vulnerability to steal a victim's cookie, by loading the `sendmail.php` script from our web server (`css.csail.mit.edu`).

Ben Bitdiddle is building a web browser, and he comes up with a plan to stop cross-site scripting attacks like the one above. His idea is to add an extra HTTP header, set by the web server, that prevents the browser from making *any* cross-origin requests from that page—including `IMG` tags, `SCRIPT` tags, forms, links, etc. If this header is set, Ben reasons that a cross-site scripting exploit like the one above will be unable to get to `sendmail.php`, and will not be able to steal the victim's cookie.

Explain how an adversary can steal a victim's cookie in Zoobar (say, running at `zoobar.com`), given a cross-site scripting vulnerability, without making cross-origin requests (i.e., bypassing Ben's plan), without network or DNS attacks, and without exploiting any other vulnerabilities.

Write javascript that saves the cookie to the user's public profile.

This page intentionally left blank.

V Privacy

9. [6 points]:

Tor uses TLS (which provides confidentiality and integrity) between onion routers, and encrypts cells traversing these routers with AES (see Figure 1). In addition, Tor checks integrity at the edges of each stream. Why is it necessary to perform end-to-end stream integrity in addition to TLS between routers?

To make sure there are no malicious routers

The manifest for the FriendTracker application is as follows:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
          package="org.siislab.tutorial.friendtracker"
          android:versionCode="1" android:versionName="1.0.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".FriendTrackerControl" android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <provider android:authorities="friends" android:name="FriendProvider"
                  android:readPermission="org.siislab.tutorial.permission.READ_FRIENDS"
                  android:writePermission="org.siislab.tutorial.permission.WRITE_FRIENDS">
        </provider>

        <service android:name="FriendTracker" android:process=":remote"
                 android:permission="org.siislab.tutorial.permission.FRIEND_SERVICE">
        </service>

        <receiver android:name="BootReceiver">
            <intent-filter>
                <action android:name="android.intent.action.BOOT_COMPLETED"></action>
            </intent-filter>
        </receiver>
    </application>

    <permission android:name="org.siislab.tutorial.permission.READ_FRIENDS"></permission>
    <permission android:name="org.siislab.tutorial.permission.WRITE_FRIENDS"></permission>
    <permission android:name="org.siislab.tutorial.permission.FRIEND_SERVICE"></permission>
    <permission android:name="org.siislab.tutorial.permission.FRIEND_SERVICE_ADD"></permission>
    <permission android:name="org.siislab.tutorial.permission.FRIEND_NEAR"
                android:label="@string/permlab_friendNear"
                android:description="@string/permdesc_friendNear"
                android:protectionLevel="dangerous"></permission>
    <permission android:name="org.siislab.tutorial.permission.BROADCAST_FRIEND_NEAR"></permission>

    <uses-permission android:name="org.siislab.tutorial.permission.READ_FRIENDS"></uses-permission>
    <uses-permission android:name="org.siislab.tutorial.permission.WRITE_FRIENDS"></uses-permission>
    <uses-permission android:name="org.siislab.tutorial.permission.FRIEND_SERVICE"></uses-permission>
    <uses-permission android:name="org.siislab.tutorial.permission.FRIEND_NEAR"></uses-permission>
    <uses-permission android:name="org.siislab.tutorial.permission.BROADCAST_FRIEND_NEAR"></uses-permis
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"></uses-permission>
    <uses-permission android:name="android.permission.READ_CONTACTS"></uses-permission>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-permission>
</manifest>
```

Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.858 Fall 2015

Quiz I

You have 80 minutes to answer the questions in this quiz. In order to receive credit you must answer the question as precisely as possible.

Some questions are harder than others, and some questions earn more points than others. You may want to skim them all through first, and attack them in the order that allows you to make the most progress.

If you find a question ambiguous, be sure to write down any assumptions you make. Be neat and legible. If we can't understand your answer, we can't give you credit!

Write your name and submission website email address on this cover sheet.

This is an open book, open notes, open laptop exam.
NO INTERNET ACCESS OR OTHER COMMUNICATION.

Please do not write in the boxes below.

I (xx/14)	II (xx/8)	III (xx/10)	IV (xx/8)	V (xx/12)	VI (xx/6)	VII (xx/6)	VIII (xx/12)	IX (xx/4)	X (xx/4)	Total (xx/84)

Name:

Submission website email address:

VIII Symbolic Execution

Recall from the EXE paper (*EXE: Automatically Generating Inputs of Death*) that, under certain circumstances, EXE forks execution in order to explore different paths (see the EXE paper's Section 2). For each of the two functions below, how many forks would each cause in total? The argument x is symbolic. Assume that the compiler does not optimize the code, and that EXE has enough time to follow all paths.

```
int
f1(int x)
{
  int z = 0;
  int i;
  for(i = 0; i < 10; i++){
    if(x == i){
      z += 1;
      break;
    }
  }
  return z;
}
```

Note the `break` inside the if statement, which causes the loop to terminate if the if condition is true.

15. [4 points]: How many forks for `f1()` above?

10

```
int
f2(int x[10])
{
  int z = 0;
  int i;
  for(i = 0; i < 10; i++){
    if(x[i] == 1){
      z += 1;
    }
  }
  return z;
}
```

Note that `x` is an array of symbolic integers, and that there is no `break` inside the if statement.

16. [4 points]: How many forks for `f2()` above?

1023 !

Suppose you have a server with a secret (in `theSecret`) that it should only reveal under certain circumstances. Your server contains this code that calls `authCheck()` to decide if the client input implies that it's OK to reveal the secret, and (if yes) it calls `emit()` to send the secret to the client.

```
if(authCheck(input))
  emit(theSecret);
```

The `input` variable holds a complete HTTP request (URL, cookies, and other HTTP headers). You are worried that there might be logical bugs in `authCheck()` that cause it to return true in situations where theSecret should not be revealed.

17. [4 points]: Outline how you could use EXE to help you gain confidence that `authCheck()` works correctly.

add asserts and run EXE

Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.858 Fall 2014

Quiz I

You have 80 minutes to answer the questions in this quiz. In order to receive credit you must answer the question as precisely as possible.

Some questions are harder than others, and some questions earn more points than others. You may want to skim them all through first, and attack them in the order that allows you to make the most progress.

If you find a question ambiguous, be sure to write down any assumptions you make. Be neat and legible. If we can't understand your answer, we can't give you credit!

Write your name and submission website email address on this cover sheet.

This is an open book, open notes, open laptop exam.
NO INTERNET ACCESS OR OTHER COMMUNICATION.

Please do not write in the boxes below.

I (xx/12)	II (xx/14)	III (xx/14)	IV (xx/12)	V (xx/6)	VI (xx/22)	VII (xx/14)	VIII (xx/6)	Total (xx/100)

Name:

Submission website email address:

V Symbolic execution

Consider the following Python program running under the concolic execution system from lab 3, where `x` is a concolic integer that gets the value 0 on the first iteration through the loop:

```
def foo(x):
  y = x + 7
  if y > 10:
    return 0
  if y * y == 256:
    return 1
  if y == 7:
    return 2
  return 3
```

9. [6 points]:

After running `foo` with an initial value of `x=0`, what constraint would the concolic execution system send to Z3 for the second `if` statement?

NOT $(x+7) > 10$ AND $(x+7)(x+7) == 256$

VI Web security

10. [8 points]: Suppose that a user visits a mashup web page that simultaneously displays a user's favorite email site, ecommerce site, and banking site. Assume that:

- The email, ecommerce, and banking sites allow themselves to be placed in iframes (e.g., they don't prevent this using X-Frame-Options headers).
- Each of those three sites is loaded in a separate iframe that is created by the parent mashup frame.
- Each site (email, ecommerce, banking, and mashup parent) come from a different origin with respect to the same origin policy. Thus, frames cannot directly tamper with each other's state.

Describe an attack that the mashup frame can launch to steal sensitive user inputs from the email, ecommerce, or banking site.

clickjacking.

mashup frame places invisible frame ontop of banking site to steal input

11. [8 points]: Each external object in a web page has a type. That type is mentioned in the object's HTML tag (e.g., an image should have an <img> tag like <img src="http://x.com/x.gif">). An object's type is also described as a MIME type in its HTTP response (e.g., Content-type: "image/gif").

These two kinds of type specifications can mismatch due to programmer error, misconfiguration, or malice. For example, for the tag <img src="http://x.com/x.gif">, the server might return the MIME type "text/css".

Suppose that, in the case of a type mismatch, the browser uses the MIME type in the HTTP response to determine how to interpret an object. For example, if X's frame tries to load the MIME-type-less tag <img src="http://Y/file">, and Y's server returns a MIME type of "text/css", the browser will interpret the fetched object as CSS in X's frame, even though the object is embedded in X's frame as an <img> tag.

Why is this a bad security policy?

The user intended the resource to be used as an img. A compromised resource could change it's type to JS and run arbitrary code

12. [6 points]: In a SQL injection attack, attacker-controlled input is evaluated in the context of a SQL query, resulting in malicious SQL statements executing over sensitive data. Ur/Web allows web applications to directly embed SQL queries in a page; furthermore, those queries may contain information that originates from the user or an untrusted source. Why is this safe in Ur/Web?

VII Network security and Kerberos

Ben Bitdiddle is concerned about the sequence number guessing attack that Steve Bellovin described in section 2 of his paper, where an adversary can spoof a TCP connection to a server from an arbitrary source IP address, and send data on that connection.

Ben implements the following strategy that his server will use for choosing the initial sequence number ISN_s of an incoming TCP connection:

$$ISN_s = ISN_{original} \oplus IP_{src} \oplus IP_{dst} \oplus (Port_{src} || Port_{dst}) \quad (1)$$

where $\oplus$ refers to the XOR operation and $||$ refers to concatenation; the IP fields being XORed refer to the 32-bit IP addresses of the source and destination of the TCP connection; and the Port fields refer to the 16-bit source and destination ports. Assume $ISN_{original}$ increments by 64 for each new incoming connection, and initially starts at some random value.

13. [8 points]:

Explain how an adversary could still launch a sequence-number-guessing attack against Ben's server with a small number of tries.

Suppose the KDC server at MIT developed a subtle hardware problem, where the random number generator became highly predictable (e.g., it would often produce the same result when asked for a "random" number).

14. [6 points]:

How could an adversary leverage this weakness to access some user's data on a file server that uses Kerberos for authentication? Describe the minimal amount of additional access the adversary might need to mount such an attack. Assume the file server ignores IP addresses in Kerberos tickets, and that the keys of all principals were generated *before* the server developed this hardware problem.

Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.858 Fall 2013

Quiz I

You have 80 minutes to answer the questions in this quiz. In order to receive credit you must answer the question as precisely as possible.

Some questions are harder than others, and some questions earn more points than others. You may want to skim them all through first, and attack them in the order that allows you to make the most progress.

If you find a question ambiguous, be sure to write down any assumptions you make. Be neat and legible. If we can't understand your answer, we can't give you credit!

Write your name and submission website email address on this cover sheet.

This is an open book, open notes, open laptop exam.
NO INTERNET ACCESS OR OTHER COMMUNICATION.

Please do not write in the boxes below.

I (xx/11)	II (xx/7)	III (xx/10)	IV (xx/10)	V (xx/7)	VI (xx/14)	VII (xx/35)	VIII (xx/6)	Total (xx/100)

Name:

Submission website email address:

V TCP/IP

6. [7 points]: Ben Bitdiddle tries to fix the Berkeley TCP/IP implementation, described in Steve Bellovin's paper, by generating initial sequence numbers using this random number generator:

```
class RandomGenerator(object):
  def __init__(self, seed):
    self.rnd = seed

  def choose_ISN_s(self):
    isn = self.rnd
    self.rnd = hash(self.rnd)
    return isn
```

Assume that Ben's server creates a `RandomGenerator` by passing it a random `seed` value not known to the adversary, that `hash()` is a well-known hash function that is difficult to invert, and that the server calls `choose_ISN_s` to determine the ISN_s value for a newly established connection.

How can an adversary establish a connection to Ben's server from an arbitrary source IP address, without being able to snoop on all packets being sent to/from the server?

1) Open legit connection, remember ISN

2) Next isn will be hash(isn)

VI Kerberos

In a Unix Kerberos implementation, each user's tickets (including the TGT ticket for the TGS service) are stored in a per-user file in `/tmp`. The Unix permissions on this file are such that the user's UID has access to that file, but the group and others do not.

7. [7 points]: Ben Bitdiddle wants to send an email to Alyssa, and to include a copy of the Kerberos paper as an attachment, but because he stayed up late studying for this quiz, he accidentally sends his Kerberos ticket file as an attachment instead. What can Alyssa do given Ben's ticket file? Be precise.

Impersonate Ben since she has his ticket

8. [7 points]: Ben Bitdiddle stores his secret files in his Athena AFS home directory. Someone hands Alyssa P. Hacker a piece of paper with the key of the Kerberos principal of `all-night-tool.mit.edu`, which is one of the `athena.dialup.mit.edu` machines. Could Alyssa leverage her knowledge of this key to get access to Ben's secret files? Assume Alyssa *cannot* intercept network traffic. Explain either how she could do so (and in what situations this might be possible), or why it is not possible.

Use keys to log in as Ben and steal tickets from there.

VII Web security

9. [7 points]: Ben Bitdiddle sets up a private wiki for his friends, running on `scripts.mit.edu`, at `http://scripts.mit.edu/~bitdiddl/wiki`. Alyssa doesn't have an account on Ben's wiki, but wants to know what Ben and his friends are doing on that wiki. She has her own web site running on `scripts.mit.edu`, at `http://scripts.mit.edu/~alyssa/`.

How can Alyssa get a copy of a given page from Ben's wiki (say, `http://scripts.mit.edu/~bitdiddl/wiki/Secret`)?

Get Ben to visit her page, make request to the wiki and send back to Alyssa.

same-origin

Ben Bitdiddle gives up on the wiki, and decides to build a system for buying used books, hosted at `http://benbooks.mit.edu/`. His code for handling requests to `http://benbooks.mit.edu/buy` is as follows:

```
1.  def buy_handler(cookie, param):
2.    print "Content-type: text/html\r\n\r\n",
3.
4.    user = check_cookie(cookie)
5.    if user is None:
6.      print "Please log in first"
7.      return
8.
9.    book = param['book']
10.   if in_stock(book):
11.     ship_book(book, user)
12.     print "Order succeeded"
13.   else:
14.     print "Book", book, "is out of stock"
```

XSS

where the `param` argument is a dictionary of the query parameters in the HTTP request (i.e., the part of the URL after the question mark). Assume Ben's cookie handling function `check_cookie` correctly checks the cookie and returns the username of the authenticated user.

10. **[7 points]:** Is there a cross-site scripting vulnerability in Ben's code? If so, specify the line number that is vulnerable, and explain how Ben should fix it.

Yes, L14 returns unsanitized input to the user

CSRF

11. **[7 points]:** Is there a cross-site request forgery vulnerability in Ben's code? If so, specify how an adversary could exploit it.

Yes.
need to use CSRF token

12. **[7 points]:** Ben decides to port his web application to Django, and use Django's stateless CSRF protection. Explain why he should migrate his web application to a separate domain that's not under `mit.edu`.

13. [7 points]: Ben Bitdiddle moved his book store to `https://www.bitdiddlebooks.com/`, but he needs to use the popular jQuery Javascript library to make his web page interactive. He adds the following line to his web page:

```
<SCRIPT SRC="http://code.jquery.com/jquery-1.9.1.js">
```

Provide at least two reasons for why this is a bad idea from a security perspective.

- Script is loaded over http so an adversary could modify contents
- if jquery.com is compromised, ben's site will also be.

Department of Electrical Engineering and Computer Science

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

6.858 Fall 2012

Quiz II

You have 80 minutes to answer the questions in this quiz. In order to receive credit you must answer the question as precisely as possible.

Some questions are harder than others, and some questions earn more points than others. You may want to skim them all through first, and attack them in the order that allows you to make the most progress.

If you find a question ambiguous, be sure to write down any assumptions you make. Be neat and legible. If we can't understand your answer, we can't give you credit!

Write your name and submission website username (typically your Athena username) on this cover sheet.

This is an open book, open notes, open laptop exam.
NO INTERNET ACCESS OR OTHER COMMUNICATION.

Please do not write in the boxes below.

I (xx/17)	II (xx/16)	III (xx/10)	IV (xx/18)	V (xx/24)	VI (xx/12)	VII (xx/6)	Total (xx/103)

Name:

Submission website username:

I RSA timing attacks / Tor

1. [9 points]: Answer the following questions based on the paper "Remote Timing Attacks are Practical" by David Brumley and Dan Boneh.

(Circle True or False for each choice.)

A. True / False Removing access to a high-precision clock on the server running Apache would prevent the attack described in Brumley's paper from working.

B. True / False Avoiding the use of the Chinese Remainder Theorem and the associated optimizations (i.e., performing computations mod p and mod q, instead of mod n) in OpenSSL would prevent the attack described in Brumley's paper from working.

C. True / False David Brumley's attack, as described in the paper, would work almost as well if the neighborhood of n values was chosen as $g, g-1, g-2, \ldots, g-n$ (as opposed to $g, g+1, g+2, \ldots, g+n$ as in the paper).

2. [8 points]:

Alyssa wants to learn the identity of a hidden service running on Tor. She plans to set up a malicious Tor OR, set up a rendezvous point on that malicious Tor OR, and send this rendezvous point's address to the introduction point of the hidden service. Then, when the hidden service connects to the malicious rendezvous point, the malicious Tor OR will record where the connection is coming from.

Will Alyssa's plan work? Why or why not?

I RSA timing attacks / Tor

1. [9 points]: Answer the following questions based on the paper "Remote Timing Attacks are Practical" by David Brumley and Dan Boneh.

(Circle True or False for each choice.)

A. True / False Removing access to a high-precision clock on the server running Apache would prevent the attack described in Brumley's paper from working.

B. True / False Avoiding the use of the Chinese Remainder Theorem and the associated optimizations (i.e., performing computations mod p and mod q, instead of mod n) in OpenSSL would prevent the attack described in Brumley's paper from working.

C. True / False David Brumley's attack, as described in the paper, would work almost as well if the neighborhood of n values was chosen as $g, g-1, g-2, \ldots, g-n$ (as opposed to $g, g+1, g+2, \ldots, g+n$ as in the paper).

2. [8 points]:

Alyssa wants to learn the identity of a hidden service running on Tor. She plans to set up a malicious Tor OR, set up a rendezvous point on that malicious Tor OR, and send this rendezvous point's address to the introduction point of the hidden service. Then, when the hidden service connects to the malicious rendezvous point, the malicious Tor OR will record where the connection is coming from.

Will Alyssa's plan work? Why or why not?