CS 155: Spring 2019
June 7, 2019

# CS 155 Final Exam

Print your name legibly and sign and abide by the honor code written below. This exam is open book and open notes. You may use course notes and documents that you have stored on a laptop, but you may NOT use the network connection on your laptop in any way, especially not to search the web or communicate with a friend. **You have 2 hours.**

The space allocated in this printed exam indicates the length of a good correct answer. Think carefully and answer clearly and succinctly. **Do not use the back side of a page to write your answer, as we will only digitally scan the front side of every page.** You may use the back of a page for scratch work.

*The following is a statement of the Stanford University Honor Code:*

A. *The Honor Code is an undertaking of the students, individually and collectively:*

   (1) *that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;*

   (2) *that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.*

B. *The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.*

C. *While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.*

I acknowledge and accept the Honor Code.

_______________________
*(Signature)*

_______________________ *(SUNet ID)*

_______________________ *(Print your name,* legibly!*)*

□ **GRADUATING?**

| Prob | # 1 | # 2 | # 3 | # 4 | # 5 | # 6 | # 7 | Total |
|---|---|---|---|---|---|---|---|---|
| Score | | | | | | | | |
| Max | 10 | 20 | 12 | 18 | 10 | 13 | 17 | 100 |

## 1. (*10 points*) ........................................................ True or False

For each question, please write T or F in the space provided. No explanation needed.

______ (a) An HttpOnly cookie can only be read via document.cookie.

______ (b) CORS prevents an attacker from exfiltrating sensitive data like a session cookie off of a website

______ (c) A ROP attack, as described in class, is the result of a stack buffer overflow. A ROP attack cannot be caused by a buffer overflow on the heap. You may assume that an overflow on the heap cannot modify the stack.

______ (d) VPNs are an effective method for hiding your Internet browsing from law enforcement.

______ (e) TCP-based protocols can be used for orchestrating DDoS attacks

______ (f) DNSSEC protects against DNS rebinding attacks

______ (g) Static analysis is better suited to find bugs in rarely used code than dynamic analysis

______ (h) The sudo command in Linux provides security through privilege separation.

______ (i) A web site example.com that supports HTTPS should respond to a request for http://www.example.com with a redirect to https://www.example.com.

______ (j) An e-commerce site responds to a customer purchase order with a confirmation page that contains an image of a green checkmark to indicate that the transaction succeeded, or an image of a red cross to indicate that the transaction failed (one image or the other is shown). The image shown is loaded using an <IMG> tag over HTTPS. One image is 5KB and the other is 6KB. Claim: despite the use of HTTPS, a network eavesdropper at the customer's WiFi router can tell whenever a transaction succeeds or fails.

**2.** (*20 points*) ................ Questions from all over with a short answer

(a) (*4 points*) Every certificate contains a field called `CA Flag` that is set to 'true' if the public key being certified belongs to a CA and is 'false' otherwise. When the browser verifies a certificate chain for a domain, it checks that all the certificates in the chain have the `CA flag` set to 'true', except for the leaf certificate for which the `CA flag` is ignored. Describe an attack that would be possible if the browser did not do this check; that is, it did not check that the `CA flag` is set to 'true' for the certificates in the chain.

(b) (*4 points*) A cryptographic library written in C initializes its pseudorandom number generator using the following command:

```
initRNG(&getSeed())
```

where `getSeed` is a function that returns a secure random seed, and `initRNG` is a function that initializes the pseudorandom number generator using the given seed. Explain why the line of code above makes this library completely insecure.

(c) (*4 points*) Many DDoS attacks use misconfigured UDP services to amplify traffic, but require a network without egress filtering to launch the attack. Explain why we do not know which networks are the source of these DDoS amplification attacks.

(d) (*4 points*) Amy is building a web forum at secforum.com where users can use HTML to post stylized messages. The website looks like the following:

```
<html>
  <body>
    <h2>Post 1 Title</h2>
    <p>User Content</p>
    <br/>
    <h2>Post 2 Title</h2>
    <p>User Content</p>
    <br/>
    <script type="text/javascript" src="//secforum.com/jquery.js"></script>
  </body>
</html>
```

Instead of trying to filter out malicious tags from each post title and content, Amy is considering using the following Content Security Policy (CSP):

```
<meta http-equiv="Content-Security-Policy" content="script-src ’self’">
```

Will this policy prevent an attacker from exfiltrating a victim's cookie by creating a forum post with embedded Javascript? Why or why not?

(e) (*4 points*) Briefly explain how the technique discussed in class for canary extraction can be similarly used to extract the ASLR random offset for a shared library like `libc`. You may assume that the attacker has located a heap overflow in a buffer that is always allocated next to a function pointer that points to `libc`.

**3.** (*12 points*) .............................................. Google Analytics

Google Analytics (GA) is a popular cloud service that lets developers track what pages users visit on their website. Websites use Google Analytics by embedding a small snippet of Javascript from https://www.google-analytics.com. The Javascript snippet does not operate within or load any (i)frames. The developer can later view website statistics through the GA web interface.

(a) (*2 points*) Google Analytics tracks users across page loads by setting an HTTP cookie in Javascript. Suppose that https://amazon.com uses Google Analytics. What `Domain` does the Google Analytics cookie use? Why?

(b) (*4 points*) If an attacker compromised Google's infrastructure and added malicious code to the loaded Google Analytics script, could they steal Amazon's other cookies? If so, how would they access and exfiltrate the cookie, and what could Amazon do to protect their cookies from Google?

(c) (*2 points*) Does sending the *Do Not Track* header prevent Amazon from using Google Analytics to track your behavior on its website? Justify your answer.

(d) (*4 points*) Does blocking third party cookies prevent Google from tracking your actions across sites through Google Analytics *using cookies*? Why or why not?

**4.** (*18 points*) .......................... Let's Encrypt Certificate Authority

Let's Encrypt is a popular certificate authority that provides X.509 certificates for HTTPS websites. Let's Encrypt is different than many other authorities because it uses a fully automated protocol for validating that a user owns the domain they are requesting a certificate for. Let's Encrypt validates domain ownership is by providing a random challenge token that the server operator must serve over HTTP on their domain's website at a Let's Encrypt specified path (e.g., http://domain.com/.well-known/letsencrypt).

The client first connects to Let's Encrypt and requests a token through their API, deploys that token at the specified path on their web server, then requests that Let's Encrypt connect to the domain and validate the token. If Let's Encrypt can successfully retrieve the token, it will issue a certificate for the domain.

(a) (*4 points*) A malicious ISP is attempting to acquire an illegitimate certificate for google.com. How can the ISP trick Let's Encrypt into providing them a Google certificate? Assume that Let's Encrypt's DNS client is properly configured and not vulnerable to blind injection attacks. Do not assume that the ISP is on the path between Let's Encrypt and Google.

(b) (*4 points*) What can Let's Encrypt do to try to prevent the attack you described in (a)? Hint: your solution does not need to be foolproof, but must lessen the chance of the attack's success and should not greatly increase how long it takes for a site to get a certificate.

(c) *(4 points)* Suppose the Let's Encrypt's TCP implementation used to request the validation token from the web server has a bug that causes every SYN packet to use the same initial sequence number. Can an attacker exploit this to acquire a certificate for a domain they do not control? Justify your answer.

(d) *(3 points)* To lower costs, Let's Encrypt is considering leasing space in its datacenter to other users. These other customers would use the same network subnet and upstream router, but would be isolated on different ports of a modern switch. Assuming that Let's Encrypt checks the presence of HTTP challenges from this data center, how could an attacker in the data center acquire a certificate for a domain they do not control? Hint: your attack should utilize the fact that Let's Encrypt and attacker are on the same network.

(e) *(3 points)* Suppose that Let's Encrypt signs certificates using a hardware security module (HSM) that requires root access to use. The LE developers know that they should not run their web application on the public Internet as root since an attacker might find a bug in their code and would then have root access to their server. Explain how they can use `fork` and `setuid` to build a secure architecture.

**5.** (*10 points*) ........................................................ Memory games

Suppose the Linux kernel contained the following buggy system call handling code:

```
void bad_syscall_handler(void) {
  void (*fp)() = NULL;     // declare a function pointer
  (*fp)();                 // call it
}
```

(a) (*3 points*) What would happen if a user program running with regular user (i.e., non-root) privileges invoked this system call? What would it cause the kernel to do? Recall that `NULL` is interpreted as address zero. For the purpose of this question you may assume that address zero is unmapped in process virtual memory.

(b) (*3 points*) Explain how a user program as above can exploit this syscall handler to gain root privilege.
**Hint:** start by having the program use the `mmap` system call to map a new user-writeable and user-executable page at address zero in its virtual address space.

(c) (*4 points*) Design a software-only defense that ensures that kernel bugs like the one above (i.e., a call from kernel space into user space) cannot be exploited. Your solution should ensure that even if a bug like the one above exists in the kernel, the attacker cannot exploit it to gain root privilege.
**Hint:** you can modify the compiler used to compile the kernel.

**6.** (*13 points*) ........................................ Pointer authentication

Let's explore an approach to defend against function pointer corruptions. Suppose the compiler generates program code that works as follows: when the executable starts, it generates a random MAC key $k$ and stores $k$ in a dedicated register for the life of the program. Every time a function pointer `fp` is written to memory, the program computes a MAC $t = \text{MAC}(k, \texttt{fp})$, and writes $t$ to a memory location adjacent to the function pointer. Whenever the program calls the function that `fp` point to, the program first checks that the MAC $t$ is a valid MAC for the value of the function pointer, and if so, calls the function.

(a) (*3 points*) What should the program do if the MAC $t$ is not a valid MAC for the value of the function pointer being called?

(b) (*3 points*) Explain why this approach makes it harder to exploit a buffer overflow in the stack or the heap that overwrites the value of a function pointer.

(c) (*3 points*) Why is it important that the key $k$ be stored in a register? Describe an attack that would be possible if $k$ were written to memory.
**Hint:** consider a program that also has a format string vulnerability.

(d) (*4 points*) Suppose the MAC $t$ were only a single byte (eight bits). Would this be sufficient to prevent an attacker from mounting a control hijacking attack by overwriting a function pointer using a buffer overflow? If so, explain why. If not, describe how the attack would work.

## 7. (*17 points*) .................................................. Encrypted SNI

(a) (*2 points*) Briefly explain what is the purpose of the Server Name Indication (SNI) extension sent as part of the TLS `client-hello` message. Consider the case of a content distribution network (CDN) that hosts thousands of domains on a single IP address.

(b) (*3 points*) Say a client (a web browser) wants to connect to `example.com` using HTTPS. Recall that in TLS 1.3, the `example.com` server certificate is encrypted when sent to the client. This hides the site's identity, `example.com`, from a network eavesdropper, thereby improving user privacy. However, the SNI extension sent in the clear by the client completely reveals this information.

A new `client-hello` extension, called *encrypted SNI* (ESNI), replaces the SNI extension with an extension that contains: (1) a public-key encryption of the cleartext SNI data, and (2) a hash of the public key used to encrypt the SNI data. The client obtains the public key with which to encrypt the SNI data by looking up a special DNS record for `example.com` that contains this public key. The DNS lookup should be done using DNS-over-HTTPS (DoH) to ensure that the DNS query does not leak the target domain to an observer.

Suppose every domain hosted at the CDN had its own ESNI public key. That is, `example1.com` had one public key in DNS and `example2.com` had a different public key in DNS. Would this achieve the goals of ESNI? Justify your answer.

(c) (*4 points*) Given your answer to the previous part, explain how the CDN should choose the public keys for the ESNI DNS entries for the domains that it hosts.

(d) (*5 points*) As described, TLS 1.3 with ESNI provides domain privacy against a network eavesdropper, but provides no privacy against an *active* network attacker. Show that an active network attacker can record the entire `client-hello` message from the client and later replay a slightly modified `client-hello` to the server. The response from the server to the replay reveals what domain the client originally connected to.

(e) (*3 points*) The designers of ESNI took care to defend against your attack from part (d). Can you suggest how the defense works?
**Hint:** try embedding more data in the ESNI ciphertext.