CS361S: Computer Security and Privacy

1. Course Contact 2
2. Course overview, objectives, and requirements 2
3. Class attendance 2
4. Lecture recordings 4
5. Grading 3
6. Perusall 3
7. Assignments 3
8. Midterm and final exam 4
9. Policy on Academic Accommodations 4
10. Academic Integrity 5
11. Artificial intelligence 5
12. Religious holy days 5
13. Class Calendar 5

1. Course Contact

Dr. Shravan Narayan
Classes: 3:30pm to 5pm at GDC 5.302 on Tuesday, Thursday
Canvas: https://utexas.instructure.com/courses/1450957

Office Hours: At GDC 6.430, 2:30pm to 3:30pm on Wednesday, 5pm to 6pm on Thursday.
(Emailing to let me know you're coming helps. Also email me if you need alternate meeting times)
Email: shr@cs.utexas.edu (Expect a response within 48 hours)

TA: Matthew Sickler mjs7565@utexas.edu (Always cc shr@cs.utexas.edu)
TA office hours: 10:00-11:00 am on Fridays + extra office hours on Monday 12:45pm to 1:45pm on weeks that assignment are due (will be announced on Canvas), GDC room 6.416

Syllabus last updated: Aug 28th 2026

2. Course overview, objectives, and requirements

This course focuses on computer security, covering a wide range of topics on both the defensive and offensive side of this field. Among these will be systems security and exploitation (e.g., buffer overflows and return-oriented programming), sandboxing and isolation, side channels, network security, JavaScript runtime security and more. The goal of the course is to provide an appreciation of how to think adversarially with respect to computer systems as well as an appreciation of how to reason about attacks and defenses.

The formal prerequisite for CS 361S is CS 439 (or 352 or 372) or CS 439H (or 352H or 372H).

At a minimum, students are expected to have knowledge of programming with C, compiling C applications, the memory layout of C applications including stack and heap, as well as standard terms and operations from programming and compilation such as how control flow (branches, indirect function calls) work, what a program counter does etc. Comfort with writing small Python and JavaScript programs and using Linux/UNIX tools, terminals, ssh, debugging tools is also expected. Finally, a basic understanding of computer architecture and at least one CPU ISA (x86, ARM, RISC-V) is expected.

3. Class attendance

The class will use a combination of slides, notes on whiteboard and live demonstrations. The slides will be made available, but you will have to take notes for whiteboard or live demonstrations. We may consider opening the collaborative class notes option on Canvas if there is enough interest for this.

If you are unable to attend some classes, it will be your responsibility to talk to your classmates to catch up with notes etc. You are always welcome to clarify your understanding of material you missed or haven't understood in the office hours.

If you have another class that conflicts with any part of this class, or if you think you cannot attend this class regularly, you should not take this class.

4. Grading

The grading scheme is as follows:

- Perusall readings + online quizzes – 13%
- Assignments 7 or 8 – 35% (not equally weighted)
- Midterm: 25%
- Final: 27%

Each of these are explained in the next sections.

Grading scheme (Final scores maybe curved by the instructor)

- A : 100% to 94%
- A- : 94% to 90%
- B+ : 90% to 87%
- B : 87% to 84%
- B- : 84% to 80%
- C+ : 80% to 77%
- C : 77% to 74%
- C- : 74% to 70%
- D+ : 70% to 67%
- D : 67% to 64%
- D- : 64% to 61%
- F : 61% to 0%

5. Perusall

We will be using perusal to assign class readings. There will be one reading every week along with a short online quiz about the week's lectures and readings. The expectation is that you will be able to commit time every week to do the reading and quiz. There are 13 perusal readings, each worth 1 point in your final grade. This point is split equally between the reading and the quiz.

6. Assignments

There will be 7 assignments (with 2 of the assignments having two parts) for a total of 9 assignment parts. The assignments are split into easy and difficult.

Assignments will be scored according to the below metric

- Assignment 1 (GDB): 3 points
- Assignment 2A (Buff overflow): 5 points
- Assignment 2B (ROP): 5 points
- Assignment 3 (Library sandboxing): 3 points
- Assignment 4 (Web): 5 points

- Assignment 5A (QuickJS Tutorial): 3 points
- Assignment 5B (QuickJS full): 5 points
- Assignment 6 (Side channels): 3 points
- Assignment 7 (LLMs): 3 points
- Total of 35 points.

You have two late days that you may use to turn in work past the deadline over the entire semester. A late day is a contiguous 24-hour period starting at the assignment due time. Use of partial late days are always rounded up to the next full day. These late days are intended to cover your extension needs for usual circumstances: brief illness, busy with other classes, interviews, travel, extracurricular conflicts, and so on. You do not need to ask permission to use a late day

Assignments maybe assigned as individual assignments or for groups of 2. If you use late days on group assignments, both you and your partner will be charged for every late day that you use, and you both must have late days to use them.

7. Midterm and final exam

This course will have a written midterm and a written final exam to be taken in-person. During the exam, no use of electronics is permitted in the exam/hallways or anywhere else until exams are turned in. There will be no makeup exams except for UT-authorized excuses. We may solely at the discretion of instructor re-weight other components of the grade to account for missed midterms for authorized reasons. See class schedule for dates.

8. Lecture recordings

This class will be recorded but lecture recordings will only be selectively released – at instructor's discretion for valid UT-approved absences (medical leave etc.) and prior to midterm/final exams. At the instructor's discretion, recordings may be made available prior to the midterm or assignments, but this is not a guarantee. So, attending class regularly will be necessary to keep up with the material.

Each student may additionally request access for upto two recorded lectures to help catchup to any missed classes.

Class recordings, if provided, are reserved only for students in this class for educational purposes and are protected under FERPA. The recordings should not be shared outside the class in any form. Violation of this restriction by a student could lead to Student Misconduct proceedings.

9. Policy on Academic Accommodations

The university is committed to creating an accessible and inclusive learning environment consistent with university policy and federal and state law. Please let me know if you experience any barriers to learning so I can work with you to ensure you have equal opportunity to participate fully in this course. If you are a student with a disability, or think you may have a disability, and need accommodation please contact Disability and Access (D&A). Please refer to D&A's website for contact and more information: http://diversity.utexas.edu/disability/. If you are already registered with D&A , please deliver your Accommodation Letter to me as early as possible in the semester so we can discuss your approved accommodation and needs in this course.

10. Academic Integrity

Recall the Student Honor Code: "As a student of The University of Texas at Austin, I shall abide by the core values of the University and uphold academic integrity."

Students who violate University rules on academic dishonesty are subject to disciplinary penalties, including the possibility of failure in the course and/or dismissal from the University. Since such dishonesty harms the individual, all students, and the integrity of the University, policies on academic dishonesty will be strictly enforced. For further information, please visit the Student Conduct and Academic Integrity Website.

To detect instances of academic integrity violations in programming assignments we may use 3rd party software.

11. Artificial intelligence

The use of artificial intelligence tools (such as ChatGPT) as "assignments solvers" in this class is strictly prohibited. Any use of AI tools detected will always be reported as an attempt to cheat and an official violation of course policy. Use of AI tools as "search engines" is permitted.

12. Religious holy days

Religion (or lack thereof) is an important part of who we are. If a holy day observed by your religion falls during the semester and you require accommodation due to that, please let me know as soon as possible. Email is an acceptable form of communication. To guarantee accommodation around presentations or other big deadlines, I will need notice of at least two weeks. If you are unable (or forget!) to provide that notice, please contact me anyway in case I can still accommodate you.

University-required language: A student who is absent from an examination or cannot meet an assignment deadline due to the observance of a religious holy day may take the exam on an alternate day or submit the assignment up to 24 hours late without penalty, ONLY if proper notice of the planned absence has been given. Notice must be given at least 14 days prior to the classes which will be missed. For religious holy days that fall within the first two weeks of the semester, notice should be given on the first day of the semester. Notice must be personally delivered to the instructor and signed and dated by the instructor, or sent certified mail. Email notification will be accepted if received, but a student submitting email notification must receive email confirmation from the instructor.

13. Class Calendar

The course calendar is here to give you an idea. It is subject to change depending on how quickly we progress through the material. We may add, remove or replace material based on interest.

Note: Assignment start/end dates may also change. Perusall readings will be updated as we go. Pay attention to class announcements for calendar updates.

Theme	Class Contents	Assignments/Readings

Tue, Aug 25	Origins Systems Security	**Contents** History of Systems Security. **Extra resources** Multics: B2 Security Evaluation Memory Corruption Attacks The (almost) Complete History - Haroon Meer (2010) E.H. Spafford: "The Internet Worm Program: An Analysis" (1988)	
Thu, Aug 27	Stack Buffer overflows, C, x86, and GDB.	**Contents** How buffer overflows work. Quick demo on GDB usage. **Extra resources** Smashing the Stack for Fun and Profit - Aleph One (1996) – reformatted in 2017 Section 30.1 to Section 30.2.1 of Low-Level Software Security by Example - Úlfar Erlingsson, Yves Younan, and Frank Piessens (2010) GDB documentation: Debugging with GDB	**Perusall reading 1 due on Sept 1st** Smashing the Stack for Fun and Profit - Aleph One (1996) – reformatted in 2017
Tue, Sep 1	GDB cont. Shellcoding tips & tricks. Other memory safety bugs.	**Contents** Continued GDB usage. Shellcoding, Integer overflows, format string bugs **Extra resources** V. van der Veen, N. dutt-Sharma, L. Cavallaro, and H. Bos: "Memory Errors: The Past, the Present, and the Future" (2012) scut: "Exploiting Format String Vulnerabilities" (2001)	**Assignment 1 available (GDB) due on Sept 8th**
Thu, Sep 3	Other bugs, (Legacy) mitigations for memory safety	**Contents** Canaries, Data Execution Prevention, ASLR **Extra resources** T. de Raadt: "Exploit Mitigation Techniques: An Update after 10 Years" (2013)	**Perusall reading 2 due on Sept 8th** V. van der Veen, N. dutt-Sharma, L. Cavallaro, and H. Bos: "Memory Errors: The

			Past, the Present, and the Future" (2012)
Tue, Sep 8	Code reuse I	**Contents** Return-to-Libc, ROP **Extra resources** Nergal, "The advanced return-into-lib(c) exploits: PaX case study" (2001) R. Roemer, E. Buchanan, H. Shacham and S. Savage: "Return-Oriented Programming: Systems, Languages, and Applications" (2012)	**Assignment 2 available (Buffer overflow) due Sept 24th**
Thu, Sep 10	Code Reuse 2 (ROP), Control-flow integrity	**Contents** ROP, Control-flow graphs, control-flow integrity **Extra resources** M. Abadi, M. Budiu, Úlfar Erlingsson, and J. Ligatti: "Control-Flow Integrity: Principles, Implementations, and Applications" (2009) Nathan Burow, Scott A. Carr, Joseph Nash, Per Larsen, Michael Franz, Stefan Brunthaler, Mathias Payer: "Control-flow integrity: Precision, security, and performance" (2017)	**Perusall reading 3 due Sept 15th** R. Roemer, E. Buchanan, H. Shacham and S. Savage: "Return-Oriented Programming: Systems, Languages, and Applications" (2012)
Tue, Sep 15		**Contents** Control-flow integrity continued. Control-flow bending. **Extra resources** N. Carlini et al.: "Control-Flow Bending: On the Effectiveness of Control-Flow Integrity" (2015)	
Thu, Sep 17	Other memory safety attacks. Isolation and secure design.	**Contents** Temporal safety and use after free. Secure application design using processes **Extra resources** Low-Level Software Security by Example - Úlfar Erlingsson, Yves Younan, and Frank Piessens (2010) DieHarder: Securing the Heap – Gene Novark, Emery D. Berger (2003)	**Perusall reading 4 due Sept 22nd** Upto end of Section 4 of DieHarder: Securing the Heap – Gene Novark, Emery D. Berger (2003)

		N. Provos, M. Friedl, P. Honeyman Preventing privilege escalation (2003)	
Tue, Sep 22	Isolation in browsers	**Assignment 2 due.** **Contents** System calls subsets **Extra resources** Adam Barth, Collin Jackson, Charles Reis, Google Chrome Team - The security architecture of the chromium browser (2009) Maddie Stone (Project Zero): In-the-Wild Series: October 2020 0-day discovery (2021) T. Garfinkel: "Traps and Pitfalls: Practical Problems in System Call Interposition Based Security Tools" (2003)	
Thu, Sep 24	Software fault isolation (SFI) + Confused deputy attacks Part 1	**Contents** Classic SFI, RLBox **Extra resources** Shravan Narayan, Craig Disselkoen, Tal Garfinkel, Nathan Froyd, Eric Rahm, Sorin Lerner, Hovav Shacham, Deian Stefan: "Retrofitting Fine Grain Isolation in the Firefox Renderer" (2020) Chapters 1, 2, and 3 from G. Tan: "Principles and Implementation Techniques of Software-Based Fault Isolation" (2017)	**Assignment 3 available (Sandboxing) due Oct 6th** **Perusall reading 5 due Sept 29th** Adam Barth, Collin Jackson, Charles Reis, Google Chrome Team - The security architecture of the chromium browser (2009)
Tue, Sep 29	Software fault isolation (SFI) + Confused deputy attacks Part 2	**Contents** RLBox continues, Wasm, Classic SFI vs. Wasm	
Thu, Oct 1	Web introduction	**Contents** Overview, Browser Security Model, CSRF **Extra resources** J. Schwenk, M. Niemietz, and C. Mainka: "Same-Origin Policy: Evaluation in Modern Browsers" (2017)	**Perusall reading 6 due Oct 6th** Shravan Narayan, Craig Disselkoen, Tal Garfinkel, Nathan Froyd, Eric Rahm, Sorin Lerner, Hovav Shacham, Deian Stefan: "Retrofitting Fine Grain

		C. Reis, A. Moshchuk, and N. Oskov: "Site Isolation: Process Separation for Web Sites within the Browser" (2019)	Isolation in the Firefox Renderer" (2020)
Tue, Oct 6	Web intro cont., Web attacks	**Contents** CSRF, SQL Injection, XSS **Extra resources** php's SQL Injection OWASP XSS Cheat Sheet OWAP CSRF	**Assignment 4 available (Web) due Oct 18th**
Thu, Oct 8	Web attacks cont.		**Perusall reading 7 due Oct 13th** J. Schwenk, M. Niemietz, and C. Mainka: "Same-Origin Policy: Evaluation in Modern Browsers" (2017)
Tue, Oct 13	Web defenses	**Contents** CSP, Iframe sandbox, Clickjacking, CORS **Extra resources** Robust defenses for cross-site request forgery by Adam Barth Play safely in sandboxed IFrames by Mike West Content security policy by Joe Medley and Mike West Subresource integrity - W3C Working Draft	
Thu, Oct 15		**No class today.** Prepare for your midterm.	
Tue, Oct 20	Midterm	**Midterm exam (Location TBD) during class hours.** Try to come to class early to be safe!	**Midterm**
Thu, Oct 22	Attacking browsers through JavaScript	**Contents** Introduction to JavaScript JavaScript engine internals	**Assignment 5 available (QuickJS) due Nov 17th** **Perusall reading 8 due Oct 27th**

		Extra resources saelo: "The Art of Exploitation: Attacking JavaScript Engines" (2016) saelo: The Art of Exploitation: Compile Your Own Type Confusion: Exploiting Logic Bugs in JavaScript JIT Engines (2019) 0vercl0k: Introduction to SpiderMonkey exploitation (2018) saelo: JITSploitation I: A JIT Bug (2020) Markus Gaasedelen and Amy "itszn" Burnett's JavaScript engine exploitation reading list Moritz Eckert's Browser-Pwn awesome-browser-exploit The JavaScript engine vulnerability database	saelo: JITSploitation I: A JIT Bug (2020)
Tue, Oct 27	Attacking browsers through JavaScript cont.	**Contents** JavaScript engine exploitation	
Thu, Oct 29	Network introduction	**Contents** Network Intro, Network Attacks: DNS and BGP **Extra resources** DNS root servers – Cloudflare S. Son and V. Shmatikov: "The Hitchhiker's Guide to DNS Cache Poisoning" (2010) Wikipedia: Autonomous System Wikipedia: OSPF routing Wikipedia: Border Gateway Protocol Wikipedia: User Datagram Protocol Wikipedia: Transmission Control Protocol Wikipedia: Domain Name System	**Perusall reading 9 due Nov 3rd** a look back at "Security problems in the TCP/IP protocol suite" Steven Bellovin (2004)
Tue, Nov 3	Networks attacks	**Contents** IP, ARP Spoofing **Extra resources**	

		Security problems in the TCP/IP protocol suite (1989) by Steven Bellovin. And a look back at "Security problems in the TCP/IP protocol suite" (2004) at this paper. Known instances of accidental/malicious BGP hijacking SAD DNS Explained by Marek Vavruša and Nick Sullivan NAT Slipstreaming by Samy Kamkar	
Thu, Nov 5	Network defenses	**Contents** Firewalls, intrusion detection systems, honeypots **Extra resources** The Base Rate Fallacy and its implications for the difficulty of intrusion detection - Stefan Axelsson (1999) A DoS-Limiting Network Architecture - Yang, Wetherall, and Anderson (2005)	**Perusall reading 10 due Nov 10th** The Base Rate Fallacy and its implications for the difficulty of intrusion detection - Stefan Axelsson (1999)
Tue, Nov 10	Secure Channels	**Contents** TLS, Certs **Extra resources** A. Delignat-Lavaud and K. Bhargavan: "Network-based Origin Confusion Attacks against HTTPS Virtual Hosting" (2015)	
Thu, Nov 12	The vulnerability market	**Contents** Bug bounty programs, zero-day market, Tor **Extra resources** Inside the Zero Day market – Mark Dowd (2023) The Legitimate Vulnerability Market: Inside the Secretive World of 0-day Exploit Sales – Charlie Miller (2007) Tor: The Second-Generation Onion Router - Roger Dingledine, Nick Mathewson, Paul Syverson (2004)	**Perusall reading 11 due Nov 17th** The Legitimate Vulnerability Market: Inside the Secretive World of 0-day Exploit Sales – Charlie Miller (2007)

		Top changes in Tor since the 2004 design paper (Part 1) – Nick Mathewson (2012) Top changes in Tor since the 2004 design paper (Part 2) – Nick Mathewson (2012) Top changes in Tor since the 2004 design paper (Part 3) – Steven Murdoch (2012)	
Tue, Nov 17	Basic side-channels	**Contents** Memory and timing side-channels **Extra resources** Sec 3.1 of D. Gruss: Introduction to Software-Based Microarchitectural Attacks (2017) Embedded Cryptography 1 Cache attacks, CT, Spectre - Prouff, Emmanuel.; Renault, Guenael.; Rivain, Mattieu.; O'Flynn, Colin. 2026	**Assignment 6 available (Side-channels) due Dec 1st**
Thu, Nov 19	Microarchitecture Attacks	**Contents** Spectre attacks **Extra resources** D. Gruss: Introduction to Software-Based Microarchitectural Attacks (2017) J. Horn: "Reading Privileged Memory with a Side-Channel" (2018) L. Fiolhais and L. Sousa: "Transient-Execution Attacks: A Computer Architect Perspective" (2023)	**Perusall reading 12 due Dec 1st** Embedded Cryptography 1 Cache attacks, CT, Spectre - Prouff, Emmanuel.; Renault, Guenael.; Rivain, Mattieu.; O'Flynn, Colin. 2026
Tue, Nov 24		**No class today (Thanksgiving)**	
Thu, Nov 26		**No class today (Thanksgiving)**	
Tue, Dec 1	Recent developments: **Tentative Plan:** LLMs & security **Fallback plan:** hardware	**Contents** Security of LLMs, LLM-based security tools **Fallback plan:** CET, PAC, MTE, MPK and more **Extra resources** The attacker moves second: Stronger adaptive attacks bypass defenses against LLM jailbreaks	**Assignment 7 available (LLM) due Dec 8th**

	support for memory safety	and prompt injections - Milad Nasr, Nicholas Carlini, Chawin Sitawarin, Sander V. Schulhoff, Jamie Hayes, Michael Ilie, Juliette Pluto, Shuang Song, Harsh Chaudhari, Ilia Shumailov, Abhradeep Thakurta, Kai Yuanqing Xiao, Andreas Terzis, Florian Tramèr (2025) **Fallback plan resources** Security Analysis of Processor Instruction Set Architecture for Enforcing Control-Flow Integrity - Vedvyas Shanbhogue, Deepak Gupta, Ravi Sahita (2019) ERIM: Secure, Efficient In-process Isolation with Protection Keys (MPK) - Anjo Vahldiek-Oberwagner, Eslam Elnikety, Nuno O. Duarte, Michael Sammler, Peter Druschel, and Deepak Garg (2019) Memory tagging: A memory efficient design - A Partap, D Boneh (2022) Security analysis of memory tagging - Joe Bialek, Ken Johnson, Matt Miller, Tony Chen (2020)	
Thu, Dec 3	Finals review	**Contents** Finals review	**Perusall reading 13 due Dec 8th** P.A. Karger and R.R. Schell: "Multics Security Evaluation: Vulnerability Analysis" (1974)
Sat, Dec 12	**Finals**	**Final exam (Location TBD) 1:00 pm - 3:00 pm**	**Finals**