Mittuniversitetet
MID SWEDEN UNIVERSITY

The Complete Study Guide for DT145G Computer Security

Daniel Bosk*

studyguide.tex 2174 2015-01-15 09:21:24Z danbos

Contents

1 Scope and Aims

The course aims towards a good understanding for the requirements of a secure computer system. Problems such as authentication and access control; software security, such as buffer overflows; as well as operating system, library and application security mechanisms are treated in the course.

More specifically, after taking this course you should be able to:

- apply different cryptosystems and explain how these work,
- analyse the problems of authentication, access control and different solutions,
- explain how some common attacks on software works,
- analyse different operating system security mechanisms,
- analyse the functionality of different types of malware,
- explain different malware protection mechanisms,
- evaluate strengths and weaknesses of hardware-based security and full-disk encryption, as well as
- value and argue about different ethical aspects of computer security, e.g. surveillance.

2 Course Structure and Content

The main course literature is *Computer Security* by Gollmann [1]. This is complemented by *Security Engineering* by Anderson [2]. The course is taught using lectures, individual laboratory assignments, workshops ("hackathon labs"), seminars, and finally a written exam. You can find a more detailed timetable, containing lab sessions etc., in the following subsection. All assignments are numbered consecutively prefixed with an "L" for laboratory assignments and "S" for a seminar assignment. For details on the examination of these and more information about deadlines, see section 3.

The course covers applied cryptography used in computer security, e.g. uses of cryptography for code obfuscation or digital rights management; authentication mechanisms, access control, and intrusion detection; software security, e.g. buffer overruns and interaction between programs; some security mechanisms provided by operating system and hardware; and malicious software and how these utilise the above weaknesses. Finally we discuss some ethical implications for computer engineers.

2.1 Schedule

To make your reading of the course easier, you are presented with a suggested schedule in this section. You are free to follow this schedule or any schedule you make for yourself, but the deadlines, laboratory sessions, and lectures will follow this schedule. You will find a short summary of schedule in Table 1 on the next page. The detailed reading instructions for each item in the schedule can be found in the following sections.

Course Week	Work
1	Course Start/Foundations of Security Lab session L0 (monoalph)
2	Lecture on Information Theory Lecture on Cryptographic Mechanisms, Part I Lecture on Cryptographic Mechanisms, Part II Lab session L0 (monoalph)
3	Lecture on Identification and Authentication Lecture on Secure Protocols Lecture on Security Usability Lab session L1 (passwd)
4	Lecture on Access Control Lecture on Reference Monitors Lecture on Accountability and Non-Repudiation Lab session L1 (passwd) First seminar session S2 (pwdpolicies)
5	Lecture on Software Security Lecture on DRM and Trusted Computing Lecture on Side-Channels Lab session L1 (passwd), L3 (tools)
6	Hackathon session L4 (malware) Lab session L1 (passwd), L3 (tools)
7	Hackathon session L5 (drm) Lab session L1 (passwd), L3 (tools)
8	Hackathon session L6 (stacksmash) Presentation L3 Seminar session S7 (ethics)
9	Individual studying
10	First exam Final lab session L1 (passwd), L4 (malware), L5 (drm), L6 (stacksmash) Second seminar session S2 (pwdpolicies), L3 (tools), S7 (ethics)
+3 months	Second exam Final seminar session S2 (pwdpolicies), L3 (tools), S7 (ethics)
+6 months	Final exam

Table 1: A summary of the parts of the course and when they will (or should) be done. The table is adapted to taking this course on half-time study rate.

2.2 Foundations of Security

Gollmann's chapter on "Foundations of Computer Security" [1, Ch. 3] attempts at a definition of Computer Security and related terms, e.g. confidentiality, integrity, and availability, which we need for our treatment of the topic. After reading this chapter you you are encouraged to do exercises 3.2, 3.5, 3.6, 3.7 and 3.8 in [1].

Anderson also covers this in Chapter 1 of [2]. However, he treats a wider area than just *computer* security, he covers many aspects of security in different examples.

2.3 L0 Breaking a Monoalphabetic Cipher

If you do not have probability theory and statistics fresh in memory you are recommended to revise that. The text *Sannolikhetsteori* by Arnlind and Enblom [3] (in Swedish) treats this subject, sections 1 to 4 are recommended.

If you have previously take (or are currently taking) a course on cryptography, the material from that course covering classical cryptography is enough. Otherwise you are recommended to read *Introduktion till några klassiska chiffer* [4] (in Swedish) or chapter 1 in *Cryptography: Theory and Practice* by Stinson [5].

2.4 Information Theory

The area of Information Theory was founded in 1948 by Claude Shannon. It concerns information, e.g. how much information we gain by seeing certain data. It is also a measure of uncertainty in information, and has thus plenty of application in security and cryptography.

The concept of entropy, the main part of Information Theory, is treated in a few short texts: *A Primer on Information Theory and Privacy* [6] and applied in "How Unique Is Your Browser?" [7], both by Eckersley, and also in "Chapter 6: Shannon entropy" by Ueltschi [8]. This is then utilised in the text "Grundläggande lösenordsanalys" [9] (in Swedish), and "Of passwords and people: Measuring the effect of password-composition policies" [10] which treats passwords.

2.5 Cryptographic Mechanisms

To fully understand how many security mechanisms can be implemented we need cryptography. Cryptography has a central role for many security mechanisms. Chapter 5 in Anderson's *Security Engineering* [2] and Chapter 14 in Gollmann's *Computer Security* [1] cover the aspects of cryptography we need in this course.

To practice your understanding of these mechanisms it is recommended to do exercises 14.2, 14.3 and 14.7 in [1].

2.6 Identification and Authentication

Identification and authentication of principals have always been a central part of computer security. Why we want to do this, and how we can accomplish this is treated in Chapter 4 in [1].

Anderson also treats this topic (Chapter 2 in [2]), although in a wider perspective with less technical details.

When you have read this chapter you should do exercises 4.2, 4.3, 4.4 and 4.6 in [1]. (Also apply your knowledge of entropy to these exercises.)

2.7 Secure Protocols

As soon as two principals need to interact, there is need for a protocol which secures the communication, be it inside or between systems – even one principal communicating with itself in different points in time, which is the case when storing something for use at a later time.

Anderson gives an overview of this area in *Security Engineering* [2], Chapter 3 "Protocols". Gollmann has a more technically detailed treatment in Chapter 15 of *Computer Security* [1].

2.8 Security Usability

One important aspect of security, which traditionally is forgotten, is the users' weaknesses. The psychology of the human mind is therefore an important subject to discuss in the context of security. Anderson gives a short summary of the psychology of users, their strengths and weaknesses, in Chapter 2 "Usability and Psychology" in [2].

Also treated in this lecture is the ever-recurring problem of password policies. The material covering this area is the article "Of passwords and people: Measuring the effect of password-composition policies" [10] and its follow-up article "Can long passwords be secure and usable?" [11].

2.9 L1 Password Cracking and Social Engineering

Before doing this laboratory assignment you should read Chapter 2 "Usability and Psychology" and Chapter 5 "Cryptography" in Anderson's *Security Engineering* [2]. You should also read the compendium "Grundläggande lösenordsanalys" [9] and the papers "Human Selection of Mnemonic Phrase-based Passwords" [12] and "Of Passwords and People" [10]. After that you should read about some recent incidents where password databases have leaked, e.g. [13–16].

You should also read about APTs. First you should read about an incident striking the security company RSA in [17]. Then you will read a paper on different approaches to APT, "Sherlock Holmes and The Case of Advanced Persistent Threat" by Juels and Yen [18].

2.10 S2 Password Policies

First you must read Chapter 2 "Usability and Psychology" in [2]. Then, to participate in this seminar you must have read the paper "Of Passwords and People" by Komanduri et al. [10]. In this paper the authors have studied how different password policies affects users' choice of passwords.

2.11 Access Control

Once you have authenticated users you can support access control – and this is also one of the main reasons to authenticate them in the first place. Access

control aims at controlling who may access what, and how they may access it. This is treated by Chapter 5, followed by Chapters 11 and 12, in *Computer Security* [1]. You are also recommended to read Anderson's treatment of the subject, he treats this in Chapters 4, 8, and 9 in *Security Engineering* [2].

To establish your newly gained knowledge in this area, you should do exercises 5.1, 5.2, 5.5, 5.6, 5.8 and 5.9 in [1].

2.12 Reference Monitors

The area of reference monitors covers enforcing access controls, it also covers trusted computing base and enforcing access control on the lower layers in the system architecture. Gollmann treats this area in Chapter 6 of his book *Computer Security* [1].

Exercises 6.1, 6.3 and 6.5 in [1] are recommended for your learning.

2.13 Accountability and Non-Repudiation

The need for accountability has been apparent in civilisations for as long as they have existed. One of today's institutions which are most renowned for keeping accounts are banks, it is quite natural therefore that Anderson describes accountability with start in the experience from banks. He treats this subject in Chapter 10 "Banking and Book-keeping" in [2].

Gollmann also describes the Clark-Wilson Security Policy Model in Section 12.3 of his book [1]. This is a model of how to securely enforcing a security policy.

Further, Schneier and Kelsey describes a system for secure audit logs in their paper "Secure audit logs to support computer forensics" [19]. The construction described therein is a method to safely store audit logs in an untrusted machine; in the scheme, all log entries generated prior to a compromise will be impossible for the attacker to read, modify, or destroy undetectably. This is not interesting because you very probably will implement this scheme, because you will proably not. It is interesting because it is a bit counter-intuitive at first, it is an example of application of crypto mechanisms, and having seen it will help you to "keep your heads out of any boxes".

2.14 Software Security

Perhaps the part of security most people intuitively associate with security, and computer security in particular, is software security. This part of computer security treats vulnerabilities in software, e.g. possibility of buffer overruns or code injections. Gollmann treats this area in Chapter 10 of his book, *Computer Security* [1]. The recommended exercises to do after reading this material are 10.1, 10.3 and 10.4 [1].

Anderson also treats this subject—in Chapter 4.4 and Chapter 18 of [2]—albeit with less technical details.

2.15 DRM and Trusted Computing

Another aspect of security is to protect parts of the system from the system user, this is what Digital Rights Management is all about. A content owner

who only allows using his or her material in a certain way must have some means of ensuring this is enforced.

We also have the other perspective of the user being able to ensure the integrity of the computer system before use. E.g. if the user has a laptop while travelling, how can the user be sure no foreign intelligence agency inserted a modified version of the operating system during the customs inspection? Or, what about the computer left in the hotel room, perhaps the hotel aide replaced the bootloader to break your full-disk encryption?

Both of these perspectives boil down to the common need of trusted computing. This is treated in chapters 16, 18 and 22 in *Security Engineering* [2].

2.16 Side-Channels

When looking at secure communication it is easy to assume it is safe just because it is encrypted. This is not always true. All data communicated is provided with confidentiality, however, there is information left to be extracted. For instance the fact *that* two principals are communicating, *when* they are communicating, the time each operation takes to perform, etc., is not provided any confidentiality. The information possible to extract from this is what is called side-channel information.

There is another aspect of this too, namely covert channels. Covert channels are channels over which communication can take place, even with limited bandwidth, despite the prohibition of this due to the security policy.

An overview of this area is provided in Chapters 17 and 23 of [2]. An interesting paper on this topic is *RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis* [20] where the authors extract RSA keys using acoustic side-channels, i.e. they analyse the sound emitted by the electrical circuitry to find the computations done and hence derive the RSA key used.

2.17 L3 Tools of the Trade

Before starting this assignment you must have a wide grasp of the theory of security. If you do not, then you will not know of all mechanisms available. Hence you will neither know of all practicalities you will have to solve to use these as a developer.

2.18 L4 Malicous Software

To be able to do this assignment you should first read chapters 5, 7, 10 in *Computer Security* [1]. Then you should read section 21.3 in *Security Engineering* [2]. Finally you should read Ken Thompson's classic paper "Reflections on trusting trust" [21].

2.19 L5 Digital Rights Management

For this assignment you should first read chapters 3, 4, 5, 16, 18, 22 in *Security Engineering* [2]. Then you should read chapters 10, 14, 15 in *Computer Security* [1].

After reading the material given above you need to know about programming in assembler, specifically x86-64 assembler and some tools. For this you

should read "x86-64 Machine-Level Programming" by Bryant and O'Hallaron [22]. You also need to be acquainted with some tools, study the manual pages for objdump(1), as(1), and gdb(1).

2.20 L6 Smashing the Stack

To grasp this assignment you must first read chapters 4, 8, 9, 18 in *Security Engineering* [2], and then you must read chapters 5, 6, 7 (and optionally 8), 10, 11, 12, 20, in *Computer Security* [1].

After reading the material given above you need to know some assembly programming, specifically x86-64 assembler and some tools. For this you should read "x86-64 Machine-Level Programming" by Bryant and O'Hallaron [22]. You also need to be acquainted with some tools, for that reason, study the manual pages for objdump(1), as(1), and gdb(1).

Finally you should read a classic paper on stack smashing, the first paper on the matter to be precise, "Smashing the stack for fun and profit" [23].

2.21 S7 The Computer Engineer's Code of Ethics

This assignment is based on the Codes of Ethics of two engineering associations. Thus, before you start you must read "Code of Ethics" [24], "Software Engineering Code of Ethics and Professional Practice" [25], and finally "IEEE Code of Ethics" [26].

Once you have read this you should read two articles analysing Snowden's revelations about the NSA surveillance techniques. The first one is "Making Sense from Snowden" [27]. The second one is "Highlight from Making Sense of Snowden, Part II" [28].

Finally, in your favourite search engine, search for the string

"nsa exploit of the day site:www.schneier.com".

Read about a few of the NSA exploits presented there.

2.22 Final exam

The final exam will examine your knowledge from taking this course. Hence, it covers all the content given above.

3 Examination

The first assignment L0 Breaking a Monoalphabetic Cipher is graded with Pass (P) or Fail (F). This is reported as I104 in the Ladok database.

The laboratory assignments L1, L3, L4, L5, L6 are also graded Pass (P) or Fail (F). They are reported collectively to Ladok as L104, this corresponds to 3 ECTS credit points. Hence, you must pass all of them to have any result resported to Ladok.

There are two seminar assignments, S2 and S7, these are graded Pass (P) or Fail (F) and are reported to Ladok as S104. This corresponds to 1.5 ECTS credit points. Thus, you must pass both seminars to have any results reported to Ladok.

Finally, the written exam will be graded A–E for passing grades, F or Fx for failing grades. You will receive an Fx if you are very close to passing. In this case you may complement your written exam with an oral exam within a week from receiving the result. If you do not take this chance within a week you must retake the exam to pass. The exam is reported as T104 and corresponds to the final 3 ECTS credit points.

3.1 Handed-In Assignments

In general, all hand-ins in the course must be in a "passable" condition; i.e. they must be well-written, grammatically correct and without spelling errors, have citations and references according to [29] (see also [30] for a tutorial), and finally fulfil all requirements from the assignment instruction. If you hand something in which is not in this condition, you will receive an F without further comment.

All material handed-in must be created by yourself, or, in the case of group assignments, created by you or one of the group members. When you refer to or quote other texts, then you must provide a correct list of references and, in the case of quotations, the quoted text must be clearly marked as quoted. If any part of the document is plagiarised you risk being suspended from study for a predetermined time, not exceeding six months, due to disciplinary offence. If it is a group assignment, all group members will be held accountable for disciplinary offence unless it is clearly marked in the work who is responsible for the part containing the plagiarism.

If cooperation takes place without the assignment instruction explicitly allowing this, this will be regarded as a disciplinary offence with the risk of being suspended for a predetermined time, not exceeding six months. Unless otherwise stated, all assignments are to be done individually.

3.2 "What if I'm not done in time?"

The deadlines on this course are of great importance, make sure to keep these!

For seminars and presentations there will be three sessions during the course of a year, if you cannot make it to any of those you will have to return the next time the course is given; i.e. up to a year later. All of these sessions will be in the course schedule (in the Student Portal). If you miss a deadline for the preparation for a seminar session, then you have to go for the next seminar even if the first seminar has not passed yet.

Written assignments are graded once during the course, most often shortly after the deadline of the assignment. After the course you are offered two more attempts within a year. In total you have three chances for having your assignments graded over the period of a year. After that you should come back the next time the course is given.

No tutoring is planned after the end of the course, i.e. after the last tutoring session scheduled in the course schedule. If you are not done with your assignments during the course and want to be guaranteed tutoring you have to reregister for the next time the course is given. Reregistration is a lower priority class of applicants for a course, all students applying for the course the first time have higher priority – this includes reserves too.

Thus, if you feel that you will not be done with the course on time, it is better to stop the course at an early stage. If you register a break within three

weeks of the course start, you will be in the higher priority class of applicants the next time you apply for the course. You can register such a break yourself in the Student Portal.

3.3 "What if I'm not done in time?"

The deadlines on this course are of great importance, make sure to keep these! You must have completed the introductory assignment within its deadline. If you do not do this you will be deregistered from the course and your place will be open to other students.

For seminars and presentations there will be three sessions during the course of a year, if you cannot make it to any of those you will have to return the next time the course is given; i.e. up to a year later. All of these sessions will be in the course schedule (in the Student Portal). If you miss a deadline for the preparation for a seminar session, then you have to go for the next seminar even if the first seminar has not passed yet.

Written assignments are graded once during the course, at the latest, shortly after the deadline of the assignment. After the course your are offered two more attempts within a year. In total you have three chances for having your assignments graded over the period of a year. After that you should come back the next time the course is given.

No tutoring is planned after the end of the course, i.e. after the last tutoring session scheduled in the course schedule. If you are not done with your assignments during the course and want to be guaranteed tutoring you have to reregister for the next time the course is given. Reregistration is a lower priority class of applicants for a course, all students applying for the course the first time have higher priority – this includes reserves too.

If you by the end of the course have a majority of the assignments left undone you will have to reregister for the course the next time it is given. Whether you have completed the majority of the assignments or not is up to the teacher to decide. Talk to the teacher to see if you have to reregister or can just hand in the missing assignments.

Thus, if you feel that you will not be done with the course on time, it is better to stop the course at an early stage. If you register a break within three weeks of the course start, you will be in the higher priority class of applicants the next time you apply for the course. You can register such a break yourself in the Student Portal.

References

[1] Dieter Gollmann. *Computer Security*. 3rd ed. Chichester, West Sussex, U.K.: Wiley, 2011. ISBN: 9780470741153 (pbk.)

[2] Ross J. Anderson. *Security Engineering. A guide to building dependable distributed systems*. 2nd ed. Indianapolis, IN: Wiley, 2008. ISBN: 978-0-470-06852-6 (hbk.) URL: `http://www.cl.cam.ac.uk/~rja14/book.html`.

[3] Joakim Arnlind and Andreas Enblom. "Sannolikhetsteori". KTH:s matematiska cirkel 2007–2008, Kungliga Tekniska högskolan. 2007. URL: `http://www.math.kth.se/cirkel/2007/kompendium07.pdf`.

[4] Daniel Bosk. “En introduktion till kryptografi”. 2013. URL: http://ver.miun.se/courses/infosak/compendii/introcrypt.pdf.

[5] Douglas R. Stinson. *Cryptography : theory and practice*. 3rd ed. Boca Raton: Chapman & Hall/CRC, 2006. ISBN: 1-58488-508-4 (Hardcover).

[6] Peter Eckersley. *A Primer on Information Theory and Privacy*. Jan. 2010. URL: https://www.eff.org/deeplinks/2010/01/primer-information-theory-and-privacy.

[7] Peter Eckersley. “How Unique Is Your Browser?” In: *Privacy Enhancing Technologies*. Springer. 2010, pp. 1–18. URL: https://panopticlick.eff.org/browser-uniqueness.pdf.

[8] Daniel Ueltschi. “Chapter 6: Shannon entropy”. URL: http://www.ueltschi.org/teaching/chapShannon.pdf.

[9] Daniel Bosk. “Grundläggande lösenordsanalys”. 2013. URL: http://ver.miun.se/courses/infosakc/compendii/pwdanalysis.pdf.

[10] Saranga Komanduri, Richard Shay, Patrick Gage Kelley, Michelle L. Mazurek, Lujo Bauer, Christin Nicolas, Lorrie Faith Cranor, and Serge Egelman. “Of passwords and people: Measuring the effect of password-composition policies”. In: *CHI*. 2011. URL: http://cups.cs.cmu.edu/rshay/pubs/passwords_and_people2011.pdf.

[11] Richard Shay, Saranga Komanduri, Adam L Durity, Phillip Seyoung Huh, Michelle L Mazurek, Sean M Segreti, Blase Ur, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. “Can long passwords be secure and usable?” In: *Proceedings of the 32nd annual ACM conference on Human factors in computing systems*. ACM. 2014, pp. 2927–2936. URL: http://lorrie.cranor.org/pubs/longpass-chi2014.pdf.

[12] Cynthia Kuo, Sasha Romanosky, and Lorrie Faith Cranor. *Human Selection of Mnemonic Phrase-based Passwords*. Tech. rep. 36. Institute of Software Research, 2006. URL: http://repository.cmu.edu/isr/36/.

[13] Troy Hunt. *A brief Sony password analysis*. June 2011. URL: http://www.troyhunt.com/2011/06/brief-sony-password-analysis.html.

[14] Graham Cluley. *The worst passwords you could ever choose exposed by Yahoo Voices hack*. July 2012. URL: http://nakedsecurity.sophos.com/2012/07/13/yahoo-voices-poor-passwords/.

[15] Jon Oberheide. *Brief analysis of the Gawker password dump*. Dec. 2010. URL: https://blog.duosecurity.com/2010/12/brief-analysis-of-the-gawker-password-dump/.

[16] Nik Cubrilovic. *RockYou Hack: From Bad to Worse*. Dec. 2009. URL: http://techcrunch.com/2009/12/14/rockyou-hack-security-myspace-facebook-passwords/.

[17] Dennis Fisher. “RSA: SecurID Attack Was Phishing Via an Excel Spreadsheet”. Apr. 2011. URL: https://threatpost.com/en_us/blogs/rsa-securid-attack-was-phishing-excel-spreadsheet-040111.

[18] Ari Juels and Ting-Fang Yen. “Sherlock Holmes and The Case of the Advanced Persistent Threat”. In: *LEET*. 2012. URL: https://www.rsa.com/rsalabs/staff/bios/ajuels/publications/SherlockHolmes.pdf.

[19] Bruce Schneier and John Kelsey. "Secure audit logs to support computer forensics". In: *ACM Transactions on Information and System Security (TISSEC)* 2.2 (1999), pp. 159–176.

[20] Daniel Genkin, Adi Shamir, and Eran Tromer. *RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis*. Tech. rep. Cryptology ePrint Archive, Report 2013/857, 2013., 2013. URL: http://eprint.iacr.org/2013/857.

[21] Ken Thompson. "Reflections on trusting trust". In: *Communications of the ACM* 27.8 (1984), pp. 761–763. URL: http://dl.acm.org/citation.cfm?id=358210.

[22] David R. Bryant Randal E. and O'Hallaron. *x86-64 Machine-Level Programming*. Sept. 2005. URL: https://www.cs.cmu.edu/~fp/courses/15213-s07/misc/asm64-handout.pdf.

[23] Aleph One. "Smashing the stack for fun and profit". In: *Phrack magazine* 7.49 (1996), pp. 14–16. URL: http://www-inst.eecs.berkeley.edu/~cs161/fa08/papers/stack_smashing.pdf.

[24] Association for Computing Machinery. *Code of Ethics: ACM Code of Ethics and Professional Conduct*. Accessed on 4 April 2014. URL: https://www.acm.org/about/code-of-ethics.

[25] Association for Computing Machinery. *Software Engineering Code of Ethics and Professional Practice*. Accessed on 4 April 2014. URL: https://www.acm.org/about/se-code.

[26] Institute of Electrical and Electronics Engineers. *IEEE Code of Ethics*. Accessed on 4 April 2014. URL: http://www.ieee.org/about/corporate/governance/p7-8.html.

[27] Susan Landau. "Making Sense from Snowden: What's Significant in the NSA Surveillance Revelations". In: *IEEE Security & Privacy* 11.4 (2013), pp. 54–63. ISSN: 1540-7993. DOI: http://dx.doi.org/10.1109/MSP.2013.90.

[28] Susan Landau. "Highlights from Making Sense of Snowden, Part II: What's Significant in the NSA Revelations". In: *IEEE Security & Privacy* 12.1 (2014), pp. 62–64. ISSN: 1540-7993. DOI: http://dx.doi.org/10.1109/MSP.2013.161.

[29] D Graffox. *IEEE Citation Reference*. Sept. 2009. URL: http://www.ieee.org/documents/ieeecitationref.pdf.

[30] Joshua M. Paiz, Elizabeth Angeli, Jodi Wagner, Elena Lawrick, Kristen Moore, Michael Anderson, Lars Soderlund, Allen Brizee, and Russell Keck. *In-Text Citations: The Basics*. Nov. 2013. URL: https://owl.english.purdue.edu/owl/owlprint/560/.