**Modern Binary  
Exploitation**

|=----------------------------------------------------------------------------=|

|=----------------------=[ Modern Binary Exploitation ]=----------------------=|

|=------------------------=[ CSCI 4968 - Spring '15 ]=------------------------=|

|=----------------------------------------------------------------------------=|

|=------------------------------=[ by RPISEC ]=-------------------------------=|

|=---------------------------=[ contact@rpis.ec ]=----------------------------=|

|=----------------------------------------------------------------------------=|

|=----------------------------------------------------------------------------=|

|=-----------------=[ Course materials available on [GitHub](http://github.com/RPISEC/MBE) ]=-----------------=|

|=----------------------------------------------------------------------------=|

+=======+======================================================================+

| Date | Class Notes |

+=======+======================================================================+

| 01/27 | --[ Syllabus and Review |

| | ----------------------------------- |

| | A quick overview of the syllabus, course, and what students can |

| | expect to learn. Some course terminology is covered along with a |

| | brief refresher of the background material required for the course. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/1/01_lecture.pdf) |

| | Course Syllabus: [Syllabus](http://security.cs.rpi.edu/courses/binexp-spring2015/Syllabus.pdf) |

| | RPISEC IRC: [getting on IRC](http://rpis.ec/irc) |

| | |

| | A more complete course rundown along with contact information can be |

| | found in the syllabus. Please be sure to at least read through it |

| | once to understand the mechanics of the course. |

| | |

| | Finally, class office hours will be held at RPISEC's hack nights! |

| | WHERE: Sage 3101 |

| | WHEN: Wednesdays, 7-10pm |

+-------+----------------------------------------------------------------------+

| 01/30 | --[ Tools and Basic Reverse Engineering |

| | ----------------------------------- |

| | We covered some of the most basic tools and their usage in reverse |

| | engineering. We covered mostly static tools today, and will go more |

| | in depth with dynamic tools as used in the typical reverse |

| | engineering workflow next class. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/2/02_lecture.pdf) |

| | Intro Crackmes: [challenges.zip](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/2/challenges.zip) |

+-------+----------------------------------------------------------------------+

| 02/03 | --[ Extended Reverse Engineering |

| | ----------------------------------- |

| | This class revolved around using IDA for static analysis in parallel |

| | with assembly level debugging in gdb & edb. We took a look at the |

| | crackmes from last class with this reversing workflow, along with |

| | the RPI & CMU bomb crackmes. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/3/03_lecture.pdf) |

| | Bomb Crackmes: [bombs.zip](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/3/bombs.zip) |

+-------+----------------------------------------------------------------------+

| 02/06 | --[ Reverse Engineering Lab |

| | ----------------------------------- |

| | Our first lab focuses on basic reverse engineering. While we won't |

| | be doing any heavy reverse engineering in this class, being able to |

| | debug at the assembly level is a necessary skill in exploit |

| | development. |

| | |

| | Labs will typically consists of three graded challenges of |

| | increasing difficulty. You are expected to complete the C problem |

| | (the easiest one) by the end of each lab period or you will take a |

| | grade penalty as specified in the syllabus. The rest are typically |

| | due by the start of class exactly one week later. |

| | |

| | Due: 2/13/2015 1:59PM EST |

| | Lab: [lab1.zip](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/4/lab1.zip) |

| | |

| | Your solutions to the lab must be submitted individually to the |

| | email mbespring2015 [at] gmail.com |

+-------+----------------------------------------------------------------------+

| 02/10 | --[ Introduction to Memory Corruption |

| | ----------------------------------- |

| | Memory corruption is typically at the heart of binary exploitation |

| | and is fundamental to the course. This lecture focused on classical |

| | stack smashing and demonstrating the possible effects of meaningful |

| | corruption. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/5/04_lecture.pdf) |

| | VM: <http://www.nostarch.com/hackingCD.htm> |

| | Other Materials: [lectures/5/](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/5/) |

| | Suggested Reading: Hacking: The Art of Exploitation |

| | chapters 0x320, skim 0x330-0x342 |

| | |

| | Be sure to visit office hours if you have any questions, want to |

| | review, or want help with the reversing lab. |

+-------+----------------------------------------------------------------------+

| 02/13 | --[ Memory Corruption Lab |

| | ----------------------------------- |

| | The Warzone is an in-house wargame that we've introduced as a self |

| | contained learning environment for this course. Most of the |

| | exercises and labs will take place on this server for the remainder |

| | of the course. |

| | |

| | SSH: warzone.rpis.ec 22 |

| | Username: lab2C |

| | Password: [DISTRIBUTED IN CLASS] |

| | |

| | Due: Friday 1:59pm, February 20th |

| | |

| | levels are in /levels/lab2/ - Start with /levels/lab2/lab2C |

| | |

| | Solve the challenge, get a shell, cat /home/lab2B/.pass |

| | Then SSH into the Warzone with the account lab2B with that password. |

| | Repeat for lab2B, then lab2A, then lab2end |

| | |

| | To submit, send the passwords, a description of your exploit, and |

| | copy of your script or payload to mbespring2015+lab2 [at] gmail.com |

| | |

| | Reminder: the bonus flag does not count toward your grade |

+-------+----------------------------------------------------------------------+

| 02/20 | --[ Shellcoding |

| | ----------------------------------- |

| | Today's class introduced the concept shellcode and how it is used in |

| | classical exploitation when injecting arbitrary code into exploited |

| | processes. We'll be moving into more modern uses of shellcode as we |

| | get deeper into the course. |

| | |

| | We also cover some simple techniques to write your own shellcode, as |

| | well as some tools we use to make shellcoding a more pain free |

| | process. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/7/05_lecture.pdf) |

+-------+----------------------------------------------------------------------+

| 02/24 | --[ Shellcoding Lab |

| | ----------------------------------- |

| | Lab today will focus on your ability to use shellcode in |

| | exploitation, and your ability to write custom shellcode tailored to |

| | the constraints of a given scenario. |

| | |

| | SSH: warzone.rpis.ec 22 |

| | Username: lab3C |

| | Password: [DISTRIBUTED IN CLASS] |

| | |

| | Due: Tuesday 1:59pm, March 3rd |

| | |

| | Submit your final command/payload for each level and a short |

| | description to mbespring2015+lab3 [at] gmail.com |

| | |

| | Lab Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/8/08_lab.pdf) |

+-------+----------------------------------------------------------------------+

| 02/27 | --[ Format Strings |

| | ----------------------------------- |

| | Format string based vulnerabilities are less common nowadays, but |

| | they are an important bug class that can be tricky to exploit. |

| | |

| | This lecture covers uncontrolled format string vulnerabilities and |

| | how they can be abused to leak information or take control of a |

| | vulnerable application. |

| | |

| | We will be having a format string based lab next week. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/9/06_lecture.pdf) |

+-------+----------------------------------------------------------------------+

| 03/03 | --[ Format String Lab |

| | ----------------------------------- |

| | SSH: warzone.rpis.ec 22 |

| | Username: lab4C |

| | Password: [DISTRIBUTED IN CLASS] |

| | |

| | Due: Tuesday 1:59pm, March 10th |

| | |

| | Submit your final command/payload for each level and a short |

| | description to mbespring2015+lab4 [at] gmail.com |

| | |

| | Lab Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/10/10_lab.pdf) |

+-------+----------------------------------------------------------------------+

| 03/10 | --[ DEP and ROP |

| | ----------------------------------- |

| | Data Execution Prevention is one of the pillars of modern exploit |

| | mitigation technologies. Understanding how DEP works and how it can |

| | be bypassed is important in exploiting real world targets. |

| | |

| | This lecture covers DEP and how it can be bypassed through Return |

| | Oriented Programming. We talk about some of the tools, workflow, and |

| | constraints of writing ROP. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/11/07_lecture.pdf) |

+-------+----------------------------------------------------------------------+

| 03/13 | --[ ROP Lab |

| | ----------------------------------- |

| | SSH: warzone.rpis.ec 22 |

| | Username: lab5C |

| | Password: [DISTRIBUTED IN CLASS] |

| | |

| | Due: Friday 1:59pm, March 20th |

| | |

| | Submit your final command/payload for each level and a short |

| | description to mbespring2015+lab5 [at] gmail.com |

| | |

| | Lab Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/12/12_lab.pdf) |

+-------+----------------------------------------------------------------------+

| 03/17 | --[ Secure Systems and Game Console Exploitation |

| | ----------------------------------- |

| | As we already have a lab assigned and project one due directly after |

| | spring break, we don't want to dive into the next subject until we |

| | get back. |

| | |

| | To mix things up, we'll be taking a peek behind the curtain at some |

| | of the bugs that brought down game consoles of our generation. |

| | |

| | Game consoles are among the most secure off the shelf products |

| | consumers can buy, so it's interesting to look at the techincal |

| | aspects of the exploits and bugs that cracked them open. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/13/08_lecture.pdf) |

+-------+----------------------------------------------------------------------+

| 03/20 | --[ Project One Lab |

| | ----------------------------------- |

| | Being the Friday of spring break, we will be hosting a lab period |

| | for project one. If anyone wants help or has questions, they're |

| | encouraged to ask for any final help before we move into break. |

| | |

| | We have also released a grading rubric for the first project in case |

| | you would like some guidance on what to turn in. |

| | |

| | Project One Rubric: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/14/ProjectOneRubric.pdf) |

| | |

| | Due: Tuesday 1:59pm, March 31st |

| | |

| | You must submit your writeup as either a link to a blog post, or as |

| | a PDF equivalent. Here is a sample CTF [Writeup](http://gaasedelen.blogspot.com/2014/11/landing-agres-links-500-csaw-ctf-2014.html) that we would expect |

| | your project writeup to resemble. |

| | |

| | Submit your final writeup links and/or work to |

| | mbespring2015+project1 [at] gmail.com |

+-------+----------------------------------------------------------------------+

| 03/31 | --[ Address Space Layout Randomization |

| | ----------------------------------- |

| | ASLR is the second big pillar in modern exploit mitigation |

| | technologies. It's designed to mitigate exploits that rely on |

| | hardcoded code/stack/heap addresses by randomizing the layout of |

| | memory for every execution. |

| | |

| | This lecture covers ASLR and how it can be bypassed through info |

| | leaks, partial overwrites, bruteforcing, and utilizing your crash |

| | state. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/15/09_lecture.pdf) |

+-------+----------------------------------------------------------------------+

| 04/03 | --[ ASLR Lab |

| | ----------------------------------- |

| | The labs are harder this week with the bugs being less trivial to |

| | spot in source and more tricky to exploit. The challenges will also |

| | have more scenario and context to make them feel a bit more real. |

| | |

| | To mix things up even more, lab6B is a remote exploitation |

| | challenge. You can expect more remote exploitation challenges as we |

| | move into the final few lab sets in the course. |

| | |

| | SSH: warzone.rpis.ec 22 |

| | Username: lab6C |

| | Password: [DISTRIBUTED IN CLASS] |

| | |

| | Because of the delay with getting lab6B up, we have extended the due |

| | time till MIDNIGHT Friday rather than by class. |

| | |

| | Due: Friday 11:59pm, April 10th |

| | |

| | Submit your final command/payload for each level and a short |

| | description to mbespring2015+lab6 [at] gmail.com |

+-------+----------------------------------------------------------------------+

| 04/07 | --[ Heap Exploitation |

| | ----------------------------------- |

| | Many exploits found in the wild today likely touch on the heap in |

| | some form. As stack based memory corruption has grown harder to |

| | utilize, the bug hunt has continued into the heap space and brought |

| | rise to new classes of vulnerabilities and techniques. |

| | |

| | This lecture covers how the heap can be abused in exploitation |

| | through heap based overflows, use after frees, heap spraying, and a |

| | brief mention of allocator metadata corruption. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/17/10_lecture.pdf) |

+-------+----------------------------------------------------------------------+

| 04/10 | --[ Heap Exploitation Lab |

| | ----------------------------------- |

| | There are only two lab challenges this week. Lab 7A is a bit tricky |

| | and will be a remote challenge. |

| | |

| | SSH: warzone.rpis.ec 22 |

| | Username: lab7C |

| | Password: [DISTRIBUTED IN CLASS] |

| | |

| | Due: Friday 1:59pm, April 17th |

| | |

| | Submit your final command/payload for each level and a short |

| | description to mbespring2015+lab7 [at] gmail.com |

+-------+----------------------------------------------------------------------+

| 04/14 | --[ Misc Concepts & Stack Canaries |

| | ----------------------------------- |

| | Smaller points and concepts that we meant to cover in the week |

| | before spring break. |

| | This includes a basic breakdown of common integers issues, |

| | interesting things with file descriptors, and details surrounding |

| | stack canaries. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/19/11_lecture.pdf) |

+-------+----------------------------------------------------------------------+

| 04/17 | --[ Misc & Canaries Lab |

| | ----------------------------------- |

| | Three lab challenges focusing on misc concepts and canaries. Lab 8A |

| | is a remote challenge. |

| | |

| | SSH: warzone.rpis.ec 22 |

| | Username: lab8C |

| | Password: [DISTRIBUTED IN CLASS] |

| | |

| | Due: Friday 1:59pm, April 24th |

| | |

| | Submit your final command/payload for each level and a short |

| | description to mbespring2015+lab8 [at] gmail.com |

+-------+----------------------------------------------------------------------+

| 04/21 | --[ C++ Concepts and Differences |

| | ----------------------------------- |

| | C++ adds a number of conviences that C lacks. Some of these |

| | additions help mitigate common exploitation avenues that we are used |

| | to such as string mishandling. It's harder to mess up things when |

| | you have a nice std::string instead of char \*'s being thrown around. |

| | |

| | But with C++ adding more high level structures, it also opens new |

| | attack surfaces to the application, e.g. classes+vtables. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/21/12_lecture.pdf) |

+-------+----------------------------------------------------------------------+

| 04/24 | --[ C++ Concepts Lab |

| | ----------------------------------- |

| | This week there's only two lab challenges and they will focus on |

| | exploiting C++ based binaries. Both challenges will be remote |

| | services! |

| | |

| | SSH: warzone.rpis.ec 22 |

| | Username: lab9C |

| | Password: [DISTRIBUTED IN CLASS] |

| | |

| | Due: Friday 1:59pm, May 1st |

| | |

| | Submit your final command/payload for each level and a short |

| | description to mbespring2015+lab9 [at] gmail.com |

+-------+----------------------------------------------------------------------+

| 04/28 | --[ Kernel Exploitation |

| | ----------------------------------- |

| | Kernel Exploitation is the process of attacking the operating system |

| | itself. Vulnerabilities in the Kernel can result in full takeover of |

| | a system and are among the most powerful bugs we can find. |

| | |

| | This lecture is an introduction to the world of kernel exploitation. |

| | It covers basic kernel exploitation techniques such as NULL |

| | dereferences and jump-to-userland, as well as kernel land |

| | mitigations such as mmap\_min\_addr and SMEP/SMAP. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/23/13_lecture.pdf) |

+-------+----------------------------------------------------------------------+

| 05/01 | --[ Kernel Exploitation Lab |

| | ----------------------------------- |

| | We've reached the system core, but can you get root? There are two |

| | lab challenges this week. |

| | |

| | This week we've brought a separate server online to spin up personal |

| | QEMU images for you to work in and break. Unlike the Warzone, this |

| | server will only be accessible from campus so you will need to VPN |

| | in if you're trying to work from off campus. |

| | |

| | SSH: warzone-kernel.rpis.ec 22 |

| | Username: lab10C |

| | Password: [DISTRIBUTED IN CLASS] |

| | |

| | Due: Friday 1:59pm, May 8th |

| | |

| | Submit your final command/payload for each level and a short |

| | description to mbespring2015+lab10 [at] gmail.com |

+-------+----------------------------------------------------------------------+

| 05/04 | --[ Final Project Lab |

| | ----------------------------------- |

| | Class today will be a dedicated lab session to get help on the Final |

| | Project. We'll be releasing a few hints and tips to help for those |

| | that attend. It's wise to ask any questions or ask for help if |

| | you're stuck. |

| | |

| | Checkpoint #2 is due by Friday 11:59pm, May 8th |

| | The entire project is due by Friday 1:59pm, May 15th |

| | |

| | We sent the rubric out a week or two ago via email, but here it is |

| | for anyone that missed it. |

| | |

| | Final Project Rubric: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/25/ProjectTwoRubric.pdf) |

+-------+----------------------------------------------------------------------+

| 05/08 | --[ Exploitation on 64bit, ARM, Windows |

| | ----------------------------------- |

| | This course has focused exclusively on owning 32bit linux binaries, |

| | but how does exploitation differ on AMD64? What about on ARM? or |

| | even on a different OS like Windows? |

| | |

| | This lecture explores and contrasts some of the differences found on |

| | the other popular architectures and operating systems we see around |

| | us today. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/26/14_lecture.pdf) |

+-------+----------------------------------------------------------------------+

| 05/12 | --[ Automation & The Future of Exploitation |

| | ----------------------------------- |

| | You've learned how leverage bugs to pwn binaries and bypass many of |

| | the modern mitigation technologies seen today. But how can we do |

| | these things faster, and where is the field going? |

| | |

| | This lecture talks a bit about fuzzing as well as a few more |

| | interesting and advanced tools in vulnerability research. It ends |

| | the course with a brief look into the future of binary exploitation |

| | and what to expect of the field in the near future. |

| | |

| | Lecture Slides: [PDF](http://security.cs.rpi.edu/courses/binexp-spring2015/lectures/27/15_lecture.pdf) |

+-------+----------------------------------------------------------------------+

[Creative Commons License](http://creativecommons.org/licenses/by-nc/4.0/)  
  
Modern Binary Exploitation by [RPISEC](http://rpis.ec/) is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](http://creativecommons.org/licenses/by-nc/4.0/).  
Permissions beyond the scope of this license may be available at <http://rpis.ec/contact>.