# Course Approval Form

For approval of new courses and deletions or modifications to an existing course.

[registrar.gmu.edu/facultystaff/curriculum](registrar.gmu.edu/facultystaff/curriculum)

## Action Requested:
- [x] Create new course
- [ ] Delete existing course
- [ ] Modify existing course (check all that apply)

## Course Level:
- [x] Undergraduate
- [ ] Graduate

## College/School: VSE (Volgenau School of Engineering)

## Department: ECE

## Subject Code: CFRS

## Number: 775

## Effective Term:
- [x] Fall
- [ ] Spring
- [ ] Summer

## Title:

**Current Banner (30 characters max including spaces):**

**New:** Kernel Forensics and Analysis

## Credits:
- [x] Fixed 3
- [ ] Variable

## Repeat Status:
- [x] Not Repeatable (NR)
- [ ] Repeatable within degree (RD)
- [ ] Repeatable within term (RT)

## Grade Mode:
- [x] Regular (A, B, C, etc.)
- [ ] Satisfactory/No Credit
- [ ] Special (A, B C, etc. +IP)

## Schedule Type Code(s):
- [x] Lecture (LEC)
- [ ] Lab (LAB)
- [ ] Recitation (RCT)
- [ ] Internship (INT)
- [ ] Seminar (SEM)
- [ ] Studio (STU)
- [ ] Independent Study (IND)
- [ ] Internship (INT)

## Prerequisite(s):
- CFRS 761

## Corequisite(s):

## Instructional Mode:
- [x] 100% face-to-face
- [ ] Hybrid: ≤ 50% electronically delivered
- [ ] 100% electronically delivered

## Special Instructions:

(Consult University Catalog for models)

## Catalog Copy for NEW Courses Only

**Description**: Introduces students to low level programming analysis and low level API’s. Students will learn the basics of kernel level device drivers, how to load and unload software from the kernel, modification of kernel objects, interrupt and call hooking and memory hiding techniques.

**Notes**: Please list additional information for the course.

**Indicate number of contact hours**:

- 0

**When Offered**:
- [x] Fall
- [ ] Summer
- [x] Spring

## Approval Signatures

[Signature]

Jens-Peter Kaps

4/24/14

College/School Approval

5/7/14

## For Graduate Courses Only

Graduate Council Member

Provost Office

Graduate Council Approval Date

---

For Registrar Office’s Use Only: Banner ___________________________ Catalog ___________________________  
revised 2/2/10
1. CATALOG DESCRIPTION
   (a) CFRS 775: Kernel Forensics and Analysis
   (b) Prerequisite: CFRS 761
   (c) Catalogue Description: Introduces students to low level programming analysis and low level API’s. Students will learn the basics of kernel level device drivers, how to load and unload software from the kernel, modification of kernel objects, interrupt and call hooking, and memory hiding techniques with the goal of forensic identification, analysis, and reporting on kernel modifications.

2. JUSTIFICATION
   (a) Course Objectives: This course will present students with the foundations of potential forms of digital forensic analysis involving the Windows kernel and APIs especially how unknown code reacts to and affects the kernel and APIs. Also, students will learn how the Windows kernel and APIs can be modified by unknown code and how these modifications can be identified.

   (b) Course Necessity: This course expands on CFRS 761 Reverse Engineering by providing the student with knowledge specific to the Windows kernel and how it can be forensically examined and analyzed.

   (c) Relationship to Existing Courses: This course builds upon the concepts in reverse engineering and analysis laid out in Computer Forensics course CFRS 761 Reverse Engineering. This course develops the analytical skills needed to effectively forensically analyze Windows kernel, API, and other resources.

3. APPROVAL HISTORY
   Department Date: 4/24/14
   IT&E Graduate Committee Date:

4. SCHEDULING
   The course will be offered every fall and spring semester, starting fall 2014 and every regular semester thereafter.

5. PROPOSED INSTRUCTORS
   Robert Osgood, Jim Jones, Tahir Khan, and other suitably qualified faculty

6. COURSE OUTLINE
   (a) Overview
      Week 1
      Introduction, overview and review programming technologies. Course will go over basic C structure and other basics.

      Week 2
      Overview of the various tools used for programming and reverse engineering of software. Topics to include overview of IDA, Olly, etc.

      Week 3
      VM Setup and configuration for reverse engineering of software. Students will learn the basics of various software packages such as Olly debug and IDA pro. Students will learn how to configure, and understand program flow.

      Week 4
      Disassembly – Students will learn the basics of disassembling applications and how to identify C-Code constructs in a disassembled file. In class labs covering disassembling of files.

      Week 5
      The windows API – Students will learn the basics of the Windows API, Windows Registry, and networking APIs. The foundations of the windows API will be reinforced with quizzes.

      Week 6
Lecture will cover kernel debugging and rootkits analysis. Coverage of techniques rootkits use to hide code and overview of the types of rootkits that exist. Forensics will be used to determine how rootkits modify files/registry.

Week 7

Discusses low-level software/malware behavior, such as introducing backdoors, credential stealing, privilege escalation, covering tracks. Advanced forensic topics on what artifacts are left behind after low-level techniques are used.

Week 8

Mid-Term Exam

Week 9

Introduces covert low-level software/malware launching, such as process injection and replacement, hook injection, asynchronous procedure call injection, etc.

Week 10

In class lecture and lab time which will focus on writing low level software that has the ability to avoid detection and analysis. Students will create applications that perform similar calls that malware uses to hide from the system.

Week 11

Covers anti-disassembly, anti-debugging, and anti-virtual machine techniques used by low-level software/malware. Students will see the techniques used by malware that prevent disassembly and reverse engineering. Students will learn the mechanisms in-place, how to perform them and also how to identify them. Students will perform forensics on systems to determine if anti-disassembly, anti-debugging or other techniques were used.

Week 12

Introduces hardware trojans and other forms of malicious hardware. Describes implementation techniques, triggering mechanisms, and possible actions of hardware trojans.

Week 13

Introduces various invasive and non-invasive ways of detecting hardware trojans, design for hardware trust, and countermeasures against malicious hardware, such as data guards, replication, and fragmentation.

Weeks 14

Final Presentations: Students will present their reports. Additional review/questions

Week 15

Final Exam

(b) Required Reading and Reference Material

Required Text:


Recommended Text:


(c) Student Evaluation Criteria

(20%) Assignments
<table>
<thead>
<tr>
<th>Weight</th>
<th>Component</th>
</tr>
</thead>
<tbody>
<tr>
<td>30%</td>
<td>Quizzes</td>
</tr>
<tr>
<td>20%</td>
<td>Midterm</td>
</tr>
<tr>
<td>30%</td>
<td>Final (Forensic report and presentation)</td>
</tr>
</tbody>
</table>