SEVENTH FRAMEWORK PROGRAMME

Information & Communication Technologies
Trustworthy ICT

NETWORK OF EXCELLENCE

A European Network of Excellence in Managing Threats and Vulnerabilities in the Future Internet: *Europe for the World* [†]

Deliverable D3.5: Experiences with the Common Curriculum Implementation

Abstract: In this deliverable, we summarise our work on the common curriculum. We will discuss how it was conceived, how it has grown over time and how it is used. The common curriculum currently finds increasing use among the system security community, well beyond the consortium itself.

Contractual Date of Delivery	September 2014
Actual Date of Delivery	December 2014 (due to 3 months extension)
Deliverable Dissemination Level	Public
Editor	Herbert Bos
Contributors	All *SysSec* partners

The *SysSec* consortium consists of:

FORTH-ICS	Coordinator	Greece
Politecnico Di Milano	Principal Contractor	Italy
VU University Amsterdam	Principal Contractor	The Netherlands
Institut Eurécom	Principal Contractor	France
IICT-BAS	Principal Contractor	Bulgaria
Technical University of Vienna	Principal Contractor	Austria
Chalmers University	Principal Contractor	Sweden
TUBITAK-BILGEM	Principal Contractor	Turkey

[†] The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 257007.

Document Revisions & Quality Assurance

Internal Reviewers

1. Magnus Almgren (Chalmers)
2. Christian Platzer (TUV)

Revisions

Ver.	Date	By	Overview
1.0.0	23/12/2014	*Editor*	Edits all over: headings to Ch. 5, more context in Ch. 1, updated numbers
0.1.2	25/11/2014	#2	Changed the curriculum adaptation section
0.1.1	25/11/2014	#2	Direct edits in all chapters, spell check, partial restructure
0.1.0	24/11/2014	#1	Changes throughout the document, extensive review
0.0.4	19/11/2014	*Editor*	Conclusion Added
0.0.3	19/11/2014	*Editor*	10K Students Initiative Added
0.0.2	13/07/2014	*Editor*	Chapters committed.
0.0.1	08/07/2014	*Editor*	Outline and preliminary article selection complete.
0.0.0	15/06/2014	*Editor*	First outline of document.

Contents

List of Figures

1 Introduction

The EU/FP7 Syssec project had as its main goal the creation of a Network of Excellence in the field of Systems Security in Europe. One of the core activities of this network was the promotion of cyber security education which would help forge the next generations of researchers as well as industry workforce. We approached this goal from different angles. First, we organised a series of highly successful summerschools focusing on malware analysis and reverse engineering. Second, we organised a scholarship scheme that allowed short-term research exchanges in the area of system security. In this deliverable, we discuss the third and most durable approach to stimulate education in system security: the SysSec common curriculum. The development of a common curriculum has become a key component in realizing SysSec's ambitions to help instructors both from within the consortium and from the broader community.

Since its inception four years ago, the common curriculum has evolved and grown. "Evolved" in the sense that our original idea of providing full semester-length courses was dropped after feedback from the Industrial Advisory Board in favour of short modules, or slide sets. And then again, when we started adding more diverse material, such as practical challenges, environments and video lectures to the common curriculum, and then again, when we decided to link the material to so-called *scenarios* or *learning trails*. It has also "grown" in that more and more material on more and more topics has been added–not just by members of the consortium, but also by outsiders.

In addition to the original goal of a common curriculum in an online repository, the consortium decided to use the last months of the project to launch a new and fairly daring initiative. Rather than wait for the professors of security courses to come to us (as is mostly the case in the online curriculum), we decided to engage the students and their professors directly by means of an ambitious challenge, which we refer to as the "10,000 Students for Security" challenge, or the "10K Students" project for short. In this initiative, the consortium prepared a set of online, publicly accessible presentations about buffer overflows—targetting different levels of expertise—and committed itself to teach this module to at least 10,000 students. Given the remaining time, we do not aim to achieve this goal before the project officially ends, but it should be achieved within a year after that.

Now that the project is drawing to a close it is good to look at the status of the common curriculum, its structure and evolution—but also our plans for the future.

Before going into details, it is good to mention that we were pleasantly surprised about the *use* of the common curriculum. Our initial thought was that common curriculum would mostly benefit schools and instructors lacking a strong background in system security—a curriculum by experts for novices. In reality, we find that even experts are active users of the common curriculum. Even the members of the SysSec consortium keep adopting and refining the material in the repository. Also researchers in top universities in Europe and the US are using the material. We think this is good news, because if top researchers at top schools use the SysSec course material, this suggests that the material is of good quality.

1.1 Overall approach and requirements of the online repository

As pointed out by the Industrial Advisory Board, it is difficult to define a common curriculum in system security and expect it to be adopted widely. Requirements differ for different universities and different programs, and a one-size fits all solution does not work. At best, we could define topics that are important for a curriculum with an emphasis on systems security, but in the end, each degree program will make its own choices about what to include.

In addition, we would be faced with the complex task of deciding what program(s) and degree(s) to aim for. Should we define a curriculum for a master in pure system security? Or a master in computer security with system security components? Or a broader computer science master with a track in security? And should it be a one year master, or a two year master? Or should we aim for a bachelor program? Or a post graduate degree? Moreover, keeping a complete curriculum up to date and consistent over a longer time and at distributed locations is challenging.

Instead, the IAB suggested to make material available in small chunks: self-contained presentations, exam questions and assignments, such that instructors can very easily adopt them. It is very easy for participating universities/schools to restructure the curriculum and group the topics in the appropriate courses if the material is available in byte-size chunks.

The goal of SysSec is to create a system security community and to stimulate quality education on the topic. The common curriculum is an important means to this end, as it provides concrete course material.

From the outset, the common curriculum's primary objective was to reach the students through the *instructors*. Thus, we would support university professors and other educators. After all, better teachers and better course material should lead to better students.

In addition, after initially targetting novice instructors, we quickly and deliberately started aiming both for instructors who are well-versed in system security and novice instructors with little or no background in the area. Experienced instructors should be able to benefit from the advanced material developed by their colleagues, such as new slides about state of the art exploitation, novel challenges, or background material. However, the material may be even more useful for instructors without, say, a research degree in systems security. They may want to include system security in their courses, or perhaps set up a new course entirely. However, doing so from scratch is a daunting task. It requires a huge time investment to develop new course material in a clear way. The availability of such material in convenient modules would be of great help.

The common curriculum should therefore be easily accessible for instructors with minimal barriers to upload and/or download course material. As mentioned in D3.5, we have opted for a wiki-based environment, as wikis are common and have friendly learning curves.

The "10k Students" project is the most visible of all the online material in the common curriculum. It aims to introduce both awareness and deep knowledge of a longstanding security issue. Moreover, it should pique the students' interest in system security matters and serve as a catalyst for further study.

1.2 Outline

The components of the common curriculum are described in the following chapters. In Chapter 2, we explain the evolution of the common curriculum. In Chapter 3, we discuss the (gforge) infrastructure for the common curriculum. In Chapter 4, we give preliminary information about the curriculum's adoption. In Chapter 5, we discuss the "10K Students" project. Finally, in Chapter 6, we conclude.

2 Evolution of the common curriculum infrastructure

When SysSec started, the consortium did not have a very clear idea of what would constitute the common curriculum. The Industrial Advisory Board, however, proved very useful when it unequivocally advised against a reference curriculum or a compilation of courses. The reason was that the IAB members had experience with other attempts in these directions and they had all failed. Moreover, in parallel to SysSec, IEEE and ACM jointly published an extensive set of guidelines for a curriculum in computer science with security as a significant component [2]. Rather than duplicating such efforts, we link to the report from our common curriculum.

The IAB argued that a lengthy reference curriculum is not even practical: instructors interested in adding system security material to their course are not well served with full-length programs. Because of the interdependencies of material in a full course, it is complex to extract exactly the topics in which the instructors are interested.

Instead, the IAB suggested to make material available in small chunks, as much as possible. So, rather than full courses, the consortium was going to make available focused and more or less self-contained presentations, exam questions and assignments, such that instructors can very easily adopt them. As the common curriculum would be defined as a set of bite-sized lectures with their own structure and topics, participating universities can very easily restructure the curriculum and group the topics in the appropriate courses.

As we developed an infrastructure to accommodate this approach, we aimed for maximum flexibility and minimum complexity. Specifically, we wanted to allow any and all course material to be uploaded without requiring much prior knowledge to do so. While different alternatives were considered, the consortium opted for a protected wiki environment as offered by Gforge. Wikis are simple and powerful while keeping the entry barrier as low as possible.

At the same time, we did not want to open up all course material to everyone, for several reasons. First, we did not want students to have access to solutions without the consent of their instructors. Second, and probably more important, we wanted to avoid the situation that professors would be reluctant to upload material to avoid objections by their universities, e.g., because of copyright issues. In retrospect, the latter point turned out to be a more serious problem than we initially thought. Specifically, when asked whether they would be okay with us making the presentations available on the web directly, some contributors were uncomfortable with this, precisely because they feared that their university would not like it.

As we did not want to partition the material any more than necessary, we decided to keep everything centralized in the Common Curriculum repository (with the exception of the 10K Students Initiative, discussed later). Doing so has the additional benefit that researchers and instructors who already heard of the current repository (through presentations, mailings, and information on the web) do not need to hear about yet another place. However, any *bona fide* instructor who wants access to the repository can request it using a lightweight procedure. The infrastructure is now available as an online learning environment which contains lectures, exam questions and general background knowledge.

In the remainder of this deliverable, we will see that we have largely adhered to the idea of having small modules that can be served piece meal. Some contributors did deliver longer programs. As long as the content is good, we decided not to prevent such contributions with an eye on also lowering the threshold for contributing.

3 The common curriculum on Gforge

After surveying several online learning environments such as Blackboard and Moodle the consortium decided to use Gforge[1], as explained in deliverable D3.3. A full description of the Gforge webserver can be found here: [1]. From the beginning, all members have had access to the repository, as did all associated members. In the last year of the project, we have opened up access to the common curriculum even further. We now provide access to all *bona fide* instructors, after a very light vetting procedure (an email exchange with the common curriculum czar to verify that the person is indeed an instructor).

Wiki

All course material is currently kept in a wiki[2]-based environment which is both flexible and easy to use. Other features such as mailing lists, discussion boards, tracking and versioning systems like subversion are available and augment the possibilities of the learning environment. The main page (Figure 3.1) has remained stable over the past year, and simply directs users quickly to the wiki environment (rather than the mailing lists, forums, etc.) The only thing we need to point out is that the main page also lists recent news announcements.

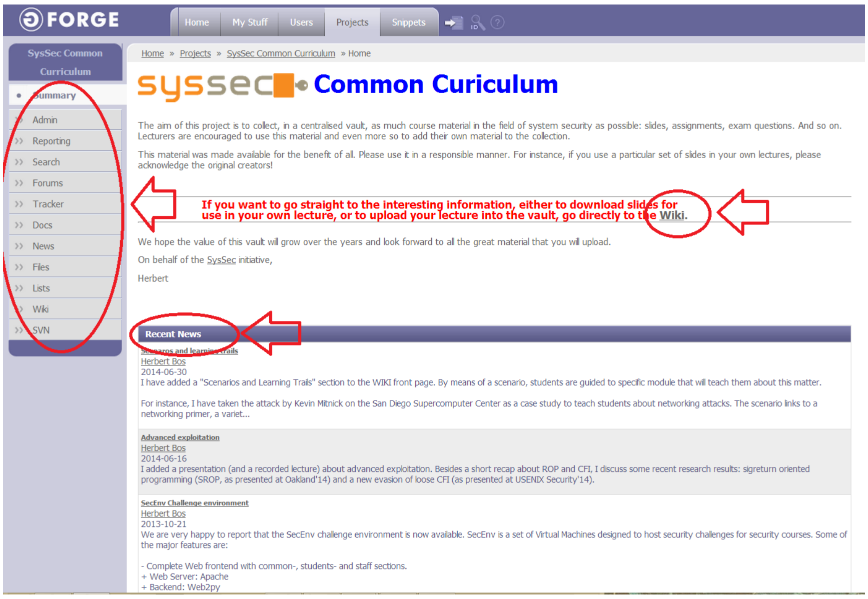
When a contributor adds material to the common curriculum infrastructure (s)he can use either the wiki or the subversion repository. In order to make this simple the wiki keeps a simple tutorial advertised on the main page of the wiki about how to add content. In practice the contributor would use the wiki to upload the initial version of the content and write

[1] `https://gforge.cs.vu.nl/gf/project/syssec_edu/`

[2] `http://en.wikipedia.org/wiki/Wiki`

the description for each topic and then use a subversion client for future updates of the content.

Figure 3.2 displays a screenshot of the current wiki main page with a link to the tutorial which explains how to add or modify content. The figure also highlights the new "Scenarios" section. We briefly discuss the main sections of the wiki.



Common Curiculum

The aim of this project is to collect, in a centralised vault, as much course material in the field of system security as possible: slides, assignments, exam questions. And so on. Lecturers are encouraged to use this material and even more so to add their own material to the collection.

This material was made available for the benefit of all. Please use it in a responsible manner. For instance, if you use a particular set of slides in your own lectures, please acknowledge the original creators!

If you want to go straight to the interesting information, either to download slides for use in your own lecture, or to upload your lecture into the vault, go directly to the Wiki.

We hope the value of this vault will grow over the years and look forward to all the great material that you will upload.

On behalf of the SysSec initiative,

Herbert

Scenaros and learning trails
Herbert Bos
2014-06-30
I have added a "Scenarios and Learning Trails" section to the WIKI front page. By means of a scenario, students are guided to specific module that will teach them about this matter.

For instance, I have taken the attack by Kevin Mitnick on the San Diego Supercomputer Center as a case study to teach students about networking attacks. The scenario links to a networking primer, a variet...

Advanced exploitation
Herbert Bos
2014-06-16
I added a presentation (and a recorded lecture) about advanced exploitation. Besides a short recap about ROP and CFI, I discuss some recent research results: sigreturn oriented programming (SROP, as presented at Oakland'14) and a new evasion of loose CFI (as presented at USENIX Security'14).

SecEnv Challenge environment
Herbert Bos
2013-10-21
We are very happy to report that the SecEnv challenge environment is now available. SecEnv is a set of Virtual Machines designed to host security challenges for security courses. Some of the major features are:

- Complete Web frontend with common-, students- and staff sections.
+ Web Server: Apache
+ Backend: Web2py

Figure 3.1: The Common Curriculum Environment main page

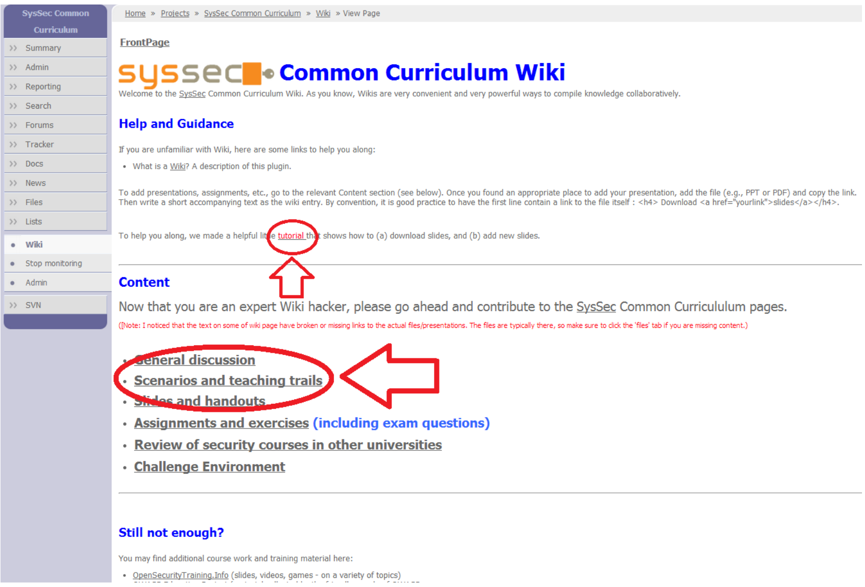
General discussion

The "General Discussion" section introduces the main idea of the common curriculum, its preference for small, self-contained modules, and guidelines for adding material.

The section can also be used for comments and discussions about the content. For instance, if participants feel that some specific content is missing, it can be added to a wish list in this section.

Scenarios

The most conspicuous change on the main wiki page in the last year has been the addition of new section for scenarios ("Scenarios and Learning Trails") to link sets of modules. As an example, consider Figure 3.3 which shows part of a scenario on network security. Using the famous attack by



FrontPage

syssec Common Curriculum Wiki

Welcome to the SysSec Common Curriculum Wiki. As you know, Wikis are very convenient and very powerful ways to compile knowledge collaboratively.

Help and Guidance

If you are unfamiliar with Wiki, here are some links to help you along:

- What is a Wiki? A description of this plugin.

To add presentations, assignments, etc., go to the relevant Content section (see below). Once you found an appropriate place to add your presentation, add the file (e.g., PPT or PDF) and copy the link. Then write a short accompanying text as the wiki entry. By convention, it is good practice to have the first line contain a link to the file itself : <h4> Download <a href="yourlink">slides</a></h4>.

To help you along, we made a helpful little tutorial that shows how to (a) download slides, and (b) add new slides.

Content

Now that you are an expert Wiki hacker, please go ahead and contribute to the SysSec Common Curricululum pages.

([Note: I noticed that the text on some of wiki page have broken or missing links to the actual files/presentations. The files are typically there, so make sure to click the 'files' tab if you are missing content.)

- General discussion
- Scenarios and teaching trails
- Slides and handouts
- Assignments and exercises (including exam questions)
- Review of security courses in other universities
- Challenge Environment

Still not enough?

You may find additional course work and training material here:

- OpenSecurityTraining.Info (slides, videos, games - on a variety of topics)

Figure 3.2: The Common Curriculum wiki main page

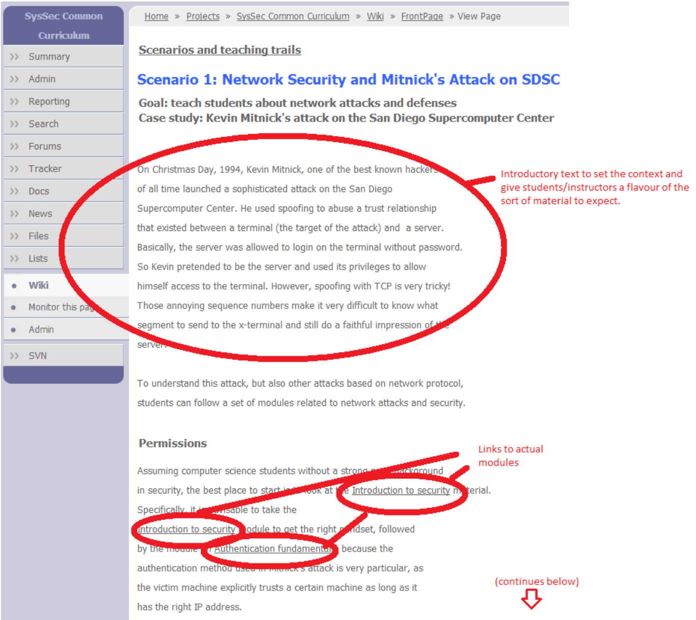
Kevin Mitnick on the San Diego Supercomputer Center as a motivating example, the scenario strings together modules that are related to the understanding of this attack. So, there will be modules about basic networking (with a security angle), modules about sniffing, modules about spoofing, etc. After following all these modules, students should be able to understand the (fairly complicated) attack, and even to re-enact a similar attack in an assignment.

Other scenarios look at other modules. For instance, a scenario on malware and reverse engineering will place more emphasis on static and dynamic analysis, assembly, IDA Pro, malware, etc.

Slides and Handouts

The bulk of the common curriculum is contained in the slides and handouts section. Since the common curriculum first opened with a single presentation about buffer overflows, this section has grown tremendously and now harbors some 15 different subcategories with some 100 presentations, screen recordings and audio clips, modules on tools like IDA Pro, handouts and online books.

An example is shown in Figure 3.4. It shows a variety of topics related to reversing in all sorts of formats. In reality, there is even more material as the topic of first SysSec summer school (of which the material is also available in the common curriculum) was similarly devoted to reverse engineering (and malware).



Scenarios and teaching trails

Scenario 1: Network Security and Mitnick's Attack on SDSC

Goal: teach students about network attacks and defenses
Case study: Kevin Mitnick's attack on the San Diego Supercomputer Center

On Christmas Day, 1994, Kevin Mitnick, one of the best known hackers of all time launched a sophisticated attack on the San Diego Supercomputer Center. He used spoofing to abuse a trust relationship that existed between a terminal (the target of the attack) and a server. Basically, the server was allowed to login on the terminal without password. So Kevin pretended to be the server and used its privileges to allow himself access to the terminal. However, spoofing with TCP is very tricky! Those annoying sequence numbers make it very difficult to know what segment to send to the x-terminal and still do a faithful impression of the server.

To understand this attack, but also other attacks based on network protocol, students can follow a set of modules related to network attacks and security.

Permissions

Assuming computer science students without a strong [illegible] background in security, the best place to start is to look at the Introduction to security material. Specifically, it is advisable to take the Introduction to security module to get the right mindset, followed by the module on Authentication fundamentals because the authentication method used in Mitnick's attack is very particular, as the victim machine explicitly trusts a certain machine as long as it has the right IP address.

Introductory text to set the context and give students/instructors a flavour of the sort of material to expect.

Links to actual modules

(continues below) ⇩

Figure 3.3: Mitnick's attack on SDSC as a scenario to link various network security modules

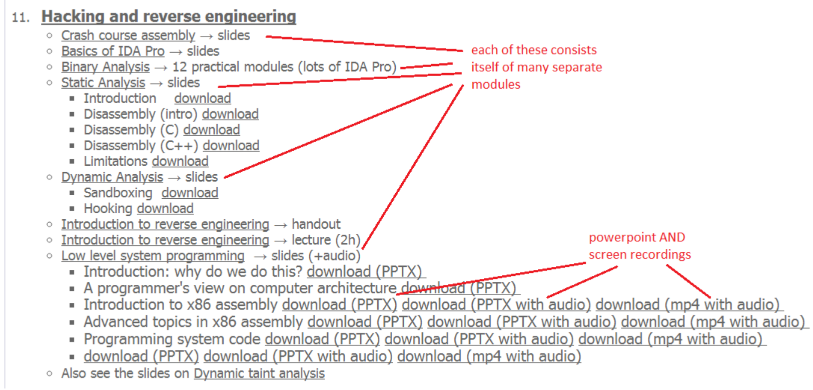
Assignments and exercises

The common curriculum also provides a section on the wiki page which contains interesting exam questions covering many topics. These are especially helpful for starting instructors who might have trouble coming up with interesting questions for the exams and can be used as is or as a starting point.

Finally, a special section of the wiki hosts a review of security courses which are taught in universities from Europe and the U.S. These are provided as suggestions of how security courses/programs could look like in terms of topics or size. These may be useful for instructors who are designing their courses or degree programs. However we believe that having the single topic (atomic) presentations is one of the key assets of the common curriculum.

Review of security courses in other universities

Originally, this section listed courses on system security at a variety of top US and EU universities with topics, teachers and links. However, if any section demonstrated the correctness of the IAB's observation about material becoming stale quickly, it is this one. Within a few months, links were no longer working, courses were changing, and the actual topics taught no



11. Hacking and reverse engineering

- Crash course assembly → slides
- Basics of IDA Pro → slides
- Binary Analysis → 12 practical modules (lots of IDA Pro)
- Static Analysis → slides
    - Introduction download
    - Disassembly (intro) download
    - Disassembly (C) download
    - Disassembly (C++) download
    - Limitations download
- Dynamic Analysis → slides
    - Sandboxing download
    - Hooking download
- Introduction to reverse engineering → handout
- Introduction to reverse engineering → lecture (2h)
- Low level system programming → slides (+audio)
    - Introduction: why do we do this? download (PPTX)
    - A programmer's view on computer architecture download (PPTX)
    - Introduction to x86 assembly download (PPTX) download (PPTX with audio) download (mp4 with audio)
    - Advanced topics in x86 assembly download (PPTX) download (PPTX with audio) download (mp4 with audio)
    - Programming system code download (PPTX) download (PPTX with audio) download (mp4 with audio)
    - download (PPTX) download (PPTX with audio) download (mp4 with audio)
- Also see the slides on Dynamic taint analysis

each of these consists itself of many separate modules

powerpoint AND screen recordings

Figure 3.4: The reverse engineering subcategory (excluding the Summer School material on reverse engineering)

longer matched the original text. While we were able to track this during the project's lifetime, it was not a future-proof solution.

To solve this, we changed the content of this section to be more future proof by linking solely to collections and reports that are decidedly stable. For instance, the "Computer Science Curricula 2013" report, published jointly by the IEEE and ACM in December of 2013 is such an example. Its expected lifetime is on the order of ten years (typically with an interim report halfway through). Admittedly, the report aims for all of computer science rather than just system security, but it has a significant section on this domain also in the Information Assurance and Security knowledge area.

In addition, we link to places like the security section on Coursera and the video collection on SecurityTube.

Challenge environment

One of the main problems with setting up a good "hands-on" security course is often the practical assignments. Frequently, instructors are quite willing to offer interesting CTF-style challenges to let students gain deep knowledge of weaknesses in systems, but doing so is not trivial. It requires an infrastructure that is fairly complex. As no such infrastructure is readily available off the shelf, organizing such labs typical demands are large investment of time and effort.

For this reason, the consortium developed the SysSec Challenge Environment, a powerful VM-based infrastructure that helps instructors set up and deploy even very complicated challenges for large groups of students. Some of the major features are:

- The environment comes with a complete web frontend with common-, students- and staff sections.
    - Web Server: Apache
    - Backend: Web2py
- We provide a virtual machine with student accounts for remote access and submission of solutions.
- The environment provides modular challenges with decoupled deployment
    - Multiple parallel challenges are possible
    - We use one VM per challenge

Like presentation and lectures, a set of these challenges is available for download. Depending on the targeted level of a course, the instructor can assemble one or more challenges to accompany the lectures.

4 Curriculum adoption

At the time of writing, the common curriculum project has about 40 active members (an increase of about 12 members per year for the last two years). Moreover, the majority (26) of these members are not from the consortium itself.

Unfortunately, even if we request members to let us know when they use material from the common curriculum, they typically do not do so and we find out only anecdotally. For instance, a professor at a related university recently sent an email where he mentioned in passing:

> *The material in the SysSec Common Curriculum served as useful inspiration. [... We used] your buffer overflow tutorial.*

As we have more leverage in the consortium itself, we made an inventory of the adoption of the common curriculum by the consortium members. Here, the case is clearer, as all the academic partners have made use of the repository for their courses.

4.1 Adoption by the partners

The adoption of the common curriculum varies from partner to partner. Often, the instructors pick up just single modules, or a few key slides here and there, which is exactly in line with the curriculum's objectives. Sometimes, entire practicals or sets of modules are adopted.

Frequently, the adoption of the SysSec material has lead to profound changes in the original courses. For instance, VU adopted the SysSec Challenge Environment for a large security challenge that involved many schools in the area of Amsterdam. Next, it decided to adopt the environment in a completely revised bachelor course. It now considers dropping all of its own challenge environment in favor of the SysSec one.

There is reason to believe that the usefulness of the common curriculum will continue to grow. The number of members in the common curriculum is growing steadily. This is encouraging, since we have only recently started releasing the common curriculum to instructors outside the consortium and its associated partners. In June 2014, we presented the common curriculum on an Intel-organized event on security education, in Leuven, Belgium which raised even more interest. Moreover, the repository will only become more useful as more material is added. Right now, we have already some 100 items in the slides and handouts section and we still grow. All of these reasons made us decide to keep supporting the common curriculum beyond the lifetime of the project and we expect that it will become an important resource for cyber security instructors in the future.

One of the associated partners explicitly mentioned that he borrowed so heavily from the SysSec slides that he no longer has a set of slides that is really his own:

> *I've been actually using the slides on the sysec wiki when possible and also borrowing from the book I'm using, so I'm not sure I really have a slide deck that is entirely my own.*

4.2 Changes in the curriculum of SysSec partners

We now summarize the impact of the common curriculum on the security courses at each of the academic partners in the SysSec consortium.

- The usefulness of the common curriculum is clearly visible within the consortium itself. For instance, Politecnico di Milano in 2013/2014 completely overhauled the materials and the challenges used in the "Computer Security" course. Many of the materials offered in the Common Curriculum were either adapted, or used as inspiration for the realization of the new coursework, which is going to be similarly shared on the Common Curriculum repository.

- Likewise, Eurécom re-designed its system security curriculum during the second year of the SysSec project. The traditional "*Network and System Security*" course was extended by a second semester course focused on binary analysis and computer forensics. This activity had a mutual impact on the SysSec common curriculum. From one side, the large amount of slides and material made available by other partners on Gforge was used as input to update the basic course, which cover several introductory topics ranging from web security, to malware and network security. On the other side, the development of the advanced course was a perfect opportunity to prepare new material that was then included in the common curriculum and shared with other partners.

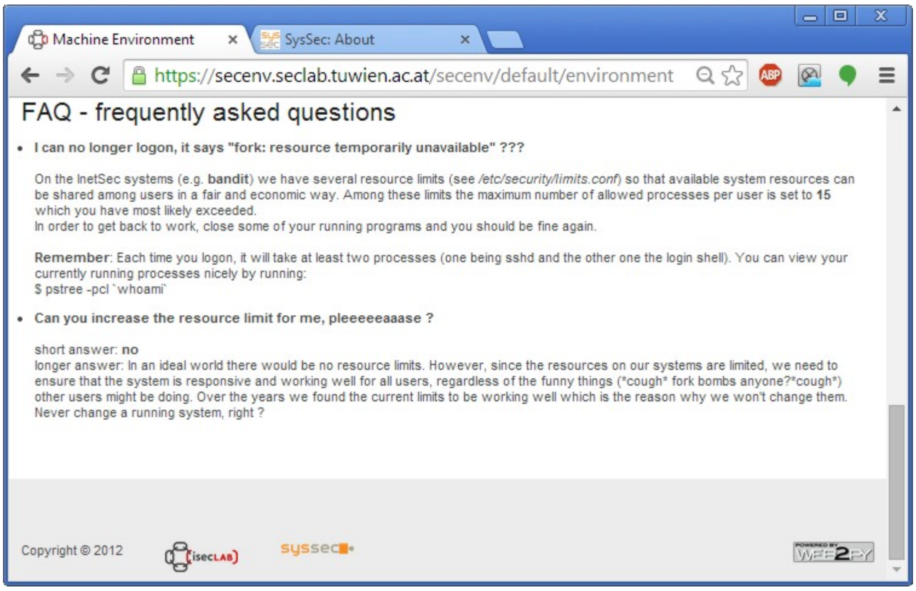
The resulting curriculum at Eurécom was very well appreciated by students, who ranked the advanced security course in the top five of the entire institute. This is also reflected in the number of students who selected the course in their study plan, which doubled in the past three years.

- The curriculum also had an impact on teaching at TUV. At the beginning of the SysSec project, security education at the Vienna University of Technology had a coverage of about 20-60 student per semester, which amounts to about 8% of all students from that semesters. Consequently, the effort put into the lectures and the computer environment was comparable low. Parallel to the start of the project, a new curriculum was introduced at TUV. This was a good opportunity to make use of the resources provided by the SysSec common curriculum. The needed teaching materials were twofold.

  - First, the lecture slides, exam questions and on-site demos needed a major overhaul. Here, the material from Gforge (see Section 3) was the most valuable asset. In fact, all the material currently presented at TUV is represented in the SysSec common curriculum. The modular structure enables the lecturers to choose which topics they want to address on a per-semester basis, which introduces an additional opportunity for dynamic lecture content.
  - The second integral part of security lectures are hands-on exercises. For this part, the challenge environment described above was installed and adopted. The SysSec-branded web frontend from TUV is shown in Figure 4.1 and can be accessed via `http://secenv.seclab.tuwien.ac.at`.

  In parallel to these developments, the new curriculum was designed to include optional security lectures in most (5/6) Bachelor curricula and some master curricula (2/6). Even though the courses are optional, they now receive 80-95% of all students, which amounts to 250-280 students from the undergraduate courses at TUV. This number is partially owed to the fact that optional means that security courses are part of a module where three out of four courses need to be chosen. Since security is perceived as interesting or even exciting, most students opt for it.

  In general, the security courses are very well-received, even though the work a student has to put into them is far higher than for the average course. The following (unmodified) student quotes were taken



FAQ - frequently asked questions

- **I can no longer logon, it says "fork: resource temporarily unavailable" ???**

  On the InetSec systems (e.g. **bandit**) we have several resource limits (see */etc/security/limits.conf*) so that available system resources can be shared among users in a fair and economic way. Among these limits the maximum number of allowed processes per user is set to **15** which you have most likely exceeded.
  In order to get back to work, close some of your running programs and you should be fine again.

  Remember: Each time you logon, it will take at least two processes (one being sshd and the other one the login shell). You can view your currently running processes nicely by running:
  $ pstree -pcl `whoami`

- **Can you increase the resource limit for me, pleeeeeaaase ?**

  short answer: **no**
  longer answer: In an ideal world there would be no resource limits. However, since the resources on our systems are limited, we need to ensure that the system is responsive and working well for all users, regardless of the funny things (*cough* fork bombs anyone?*cough*) other users might be doing. Over the years we found the current limits to be working well which is the reason why we won't change them. Never change a running system, right ?

Copyright © 2012

Figure 4.1: SysSec-branded lecture environment @ TUV

from the anonymous student evaluation at the end of 2013 for the course *Internet Security*:

> *This was the most interesting and helpful lecture I have visited so far. I've always wanted to learn a bit more about security, and this course went far beyond anything I ever hoped for! Thank you very much! It was an real fun, all the exercises were interesting, each having its own plot. Great sense of humor, broad and absolutely up-to-date topics, perfectly prepared environment for exercising.. just awesome!*
>
> *Why only two courses? :(*
>
> *Challenges were well-planned, nicely structured and partially more than dodgy!*
>
> *I really liked the challenges. In particular the innovative, up-to-date examples and the possibility for an automatic submission and grading. Finally, a lecture which is up to the minute.*

- The course material in the common curriculum also greatly helped VU to start a new introductory bachelor course on security and prompted it to add a very advanced security course (on binary and malware analysis) in the master. For the bachelor course, VU used slides from several modules in the common curriculum (a general introduction to security, some of the material on cryptography and slides on buffer overflows and memory errors). The course was evaluated very positively by the students.

The Binary and Malware analysis course has a very similar setup as the course material for the first SysSec summer school. Specifically, some of the course's challenges are extended versions of the problems in the summer school (reflecting the longer time students have available for them). Moreover, the courses uses 2 presentations from Eurécom (on static and dynamic analysis). The course is entirely hands-on and guides students through a number of increasingly complicated problems that security experts encounter in the real world when dissecting malware. The students consider the course to be very tough, but also extremely interesting and rewarding.

Besides the above two courses, VU used material of the common curriculum in its Computer and Network Security course—building on slides by TUV and Eurécom. Furthermore, it launched an initiative to teach secondary school kids about cyber security by means of the SysSec online challenge environment built by TUV mentioned earlier. The challenge environment is also available from the common curriculum site.

- Chalmers used lectures from the common curriculum from the start, such as the module explaining the buffer overflow. As the content has grown in the repository, several other modules have also been incorporated into courses in the university over the years (for example, buffer overflows, side channel attacks, mobile malware). This year, the scenarios (i.e. learning trails) will also be introduced to motivate the students and combining the lectures into a whole. Furthermore, for the next academic year (2015/2016), labs need to be upgraded for one of the introductory courses in security, and it is thus currently investigated if the Syssec challenge environment can also be utilized at Chalmers. The use of the common curriculum at Chalmers thus follows the same pattern as how we envision other institutes will use the common curriculum as it is wider released. First, only a few modules are used but over time more material, including exam questions, complete labs, etc, are utilized. This in turn leads to higher quality education for system security as a whole in Europe.

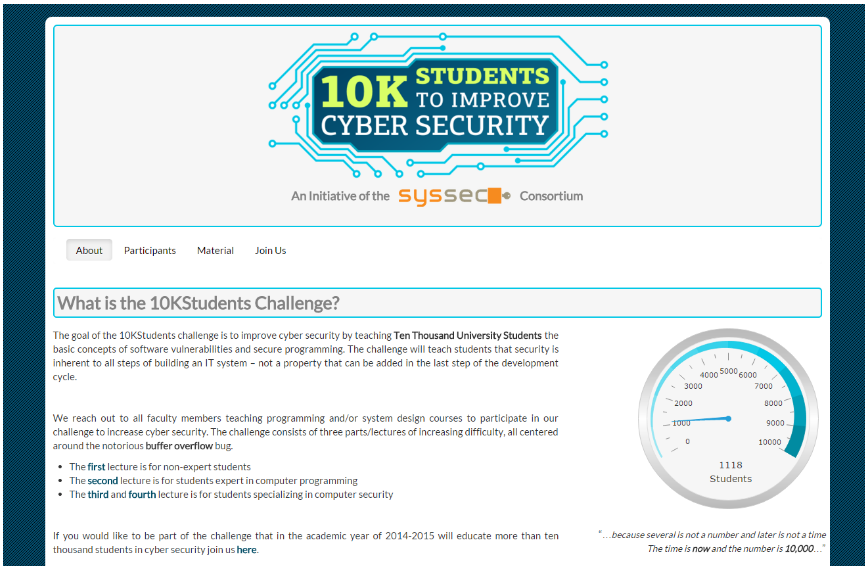
Moreover, all of the academic partners also participate in the *10K students initiative* (see next chapter). In other words, they sign up their students to solve the questions/challenges associated with the slides sets.

The positive feedback also encouraged us to extend the lifetime of the Syssec common curriculum as far as possible. With the growing number of participants it will be much easier to assemble even more interesting courses in a relatively short time.

5

The 10k Students Initiative

In addition to the original goal of a common curriculum in an online repository, the consortium decided to use the last period of the project to launch a daring new initiative: to teach 10,000 students about buffer overflows. Buffer overflows are some of the most serious and persistent security vulnerabilities that we know. They were used already in 1988, when Robert Morris Jr. launched the famous worm that brought down the Internet, and they are just as dangerous today. Unlike some other exploitable vulnerabilities, they are also relatively simple to explain.



What is the 10KStudents Challenge?

The goal of the 10KStudents challenge is to improve cyber security by teaching **Ten Thousand University Students** the basic concepts of software vulnerabilities and secure programming. The challenge will teach students that security is inherent to all steps of building an IT system – not a property that can be added in the last step of the development cycle.

We reach out to all faculty members teaching programming and/or system design courses to participate in our challenge to increase cyber security. The challenge consists of three parts/lectures of increasing difficulty, all centered around the notorious **buffer overflow** bug.

- The **first** lecture is for non-expert students
- The **second** lecture is for students expert in computer programming
- The **third** and **fourth** lecture is for students specializing in computer security

If you would like to be part of the challenge that in the academic year of 2014-2015 will educate more than ten thousand students in cyber security join us **here**.

*"...because several is not a number and later is not a time The time is **now** and the number is **10,000**..."*

Figure 5.1: The 10K Students Initiative

The consortium has committed itself to teach this module to at least 10,000 students. Given the remaining time, we do not aim to achieve this goal before the project officially ends, but it should be achieved within a year after that. The project website and all the material is available from `http://10kstudents.eu`.

5.1 Counting students

Having set the goal, we considered how to best achieve it. In other words: how would we count the number of students that participated in the challenge? Should we count everybody who happens by? Register students by means of sign-up page? People who download a presentation? Only people who pass a set of test questions? The problem is that if we make the administration too cumbersome, fewer students will be inclined to participate, but unless we make the registration thorough, it will be difficult to measure with a measure of certainty to what extent people have actually used the material.

After some deliberation, we compromised by measuring participation via the professors. Instructors interested in teaching buffer overflows in their courses can make this material available to their students and register their class with SysSec, together with the number of students participating. The courses could range from actual security courses to more generic courses on operating systems, programming, software engineering, etc.

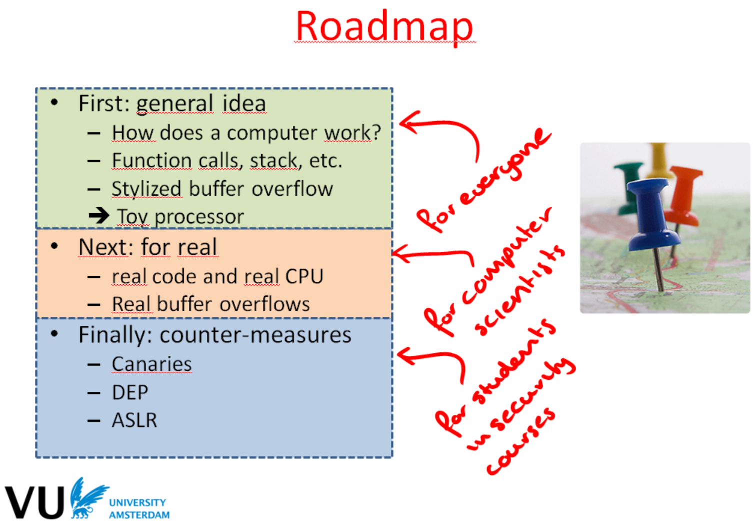
5.2 Something for everyone

Given our goal to create awareness among as diverse a student population as possible, we even want the material suitable for non-technical courses. For instance, to make students understand *why* software can be vulnerable, by teaching the general idea of a buffer overflow exploit, without all the technical details. At the same time, we also want the material to be interesting for advanced security courses at, say, the master level.

To cater to a diverse audience, we decided to split up the presentation in four different modules, where the first module provides a general overview of the problem of buffer overflows and the structure of the subsequent presentations, and each of the subsequent presentations targets a different population of students.

The second module presents the *concept* of a buffer overflow to a non-CS audience (or CS students in their first or second year of their bachelor). Specifically, the module explains how a computer executes a computer program in a stylized form. It starts with instructions, functions, the stack, and variables and shows what happens on a function call and a return from function calls, in an easy-to-comprehend, yet still realistic example.

The third module targets more advanced computer science students, but not yet security experts. It takes the same example as in the first module, but



Roadmap

- First: general idea
  - How does a computer work?
  - Function calls, stack, etc.
  - Stylized buffer overflow
  - ➔ Toy processor
- Next: for real
  - real code and real CPU
  - Real buffer overflows
- Finally: counter-measures
  - Canaries
  - DEP
  - ASLR

for everyone

for computer scientists

for students in security courses

VU UNIVERSITY AMSTERDAM

Figure 5.2: Content from the introductory presentation of the 10K Student Initiative, showing the three different technical modules

makes it more realistic, by showing real instructions, actual addresses, but especially by looking at the attack vector itself. The module shows what the attacks vector looks like, including the shellcode, the NOP sled, and possible packers.

The fourth module aims for advanced security students. In this module, we discuss counter measures, such as canaries, address space randomization, and data execution prevention. Furthermore, we explain how to counter counter measures, such as tricks to circumvent the above measures like modern return-oriented programming attacks.

Separate sets of questions for all of the three technical modules, will help students test whether they understood the material. The questions are relatively easy for the earlier modules, but quite tough and interesting for the advanced module—again with an eye on making it interesting for everyone. The instructors who register will also get the answers to the questions.

5.3 Form: easy to use, easy to understand

A final question concerned the form in which we should present the material. The consortium's aim was to make the material suitable for inclusion in any class and by whatever means. In other words, instructors are free to use the presentations directly, glean some of the slides and use them in their own course material, or refer students to the online material and challenges.

Of course, making the material usable in existing presentations of instructors simply suggests making slides available in a common form, like PDF and PowerPoint. However, the tougher problem is the explanation of these slides. Some of the material on the slides is quite complicated and perhaps not understandable without a lecturer's explanation. This is true both for the students (referred to the website for self study), and for instructors who do not have much background in exploitation (and may have a hard time understanding the finer points of the presentations). For this reason, we decided to make simple screen recordings with audio and video to "teach" the lecture material in a MOOC-like fashion.

5.4 Conclusions and outlook

The 10k students initiative, serves both as an eye-catcher for SysSec's common curriculum, and as a concrete set of modules to teach buffer overflows to students of different backgrounds. Moreover, for non expert audiences, it helps create awareness, and underlines the importance of system-level security as opposed to traditional "Alice and Bob" style cryptography.

Given the success of the initiative, we hope that it will give rise to similar challenges on other systems security topics. For instance, we can imagine challenges targetting other vulnerabilities (like dangling pointers), exploitation, and return oriented programming, but also "new" defenses like control flow integrity (CFI) and ShadowStacks. In that sense, the challenge is not just meant to teach or even to catch attention, but ideally also to inspire.

6 Conclusions

The SysSec Common Curriculum has grown from humble beginnings (three modules on a simple Wiki) to a valuable repository of all sorts of course material on system security, and now includes presentations, scenarios to tie together different modules, exam questions, and practicals. In our opinion, it is unquestionably a useful aid for instructors to set up or improve courses in this field. Moreover, the more material is added, the more useful the common curriculum becomes.

This is reflected also by the growing external interest. Where the users in the first years were mostly from the consortium, or the consortium's close circle of friends and collaborators, we now receive participation requests from instructors we did not even know and who heard about the initiative and want to make use of it.

We expect that the 10k students initiative will generate an additional boost in interest in the common curriculum. The initiative is already picked up by well-known professors in the field who respond very enthusiastically, even though we have not really started advertising it aggressively.

In our opinion, the common curriculum is well on its way to become an important focal point for online material about systems security and a valuable trove of course material in the field for instructors worldwide. Since we consider it to be important for ourselves also, we will keep supporting the common curriculum even beyond the lifetime of the SysSec project.

Bibliography

[1] Gforge Collaborative Development Environment. http://gforge.com.

[2] ACM and I. J. T. F. on Computing Curricula. Computer science curricula 2013 – curriculum guidelines for undergraduate degree programs in computer science. http://www.acm.org/education/CS2013-final-report.pdf, December 2013.