

FIRST EduC

FIRST Education & Training Committee

Are you ready for Malta?

Jose Nazario lays out the program committee's vision to reintroduce training opportunities in Malta. We are especially excited by the Technical Foundations track, the DRG Challenge, and the Open Exercise.

Page 2



Introduction

The EduC is pleased to present this second and final **pilot newsletter**. Yes, this is the last newsletter the EduC has on our agenda. In the future, we will be supporting the **FIRST Times**. We will provide articles relating to:

- Training Course Reviews
- Upcoming Training Events
- Materials for Creating Presentations
- Training / Education Content

We discovered that with proper planning the time required to collect and edit the articles is quite manageable. However, the time required to layout the newsletter is not.

In this edition we open up with a message from Jose Nazario, Program Committee Chair for Malta, on how they are addressing training opportunities during the conference. We also have a reading list, member review, update on EduC, update on Transits II, four reasonable practices for software development, and resources to create a security awareness presentation for children.

In This Issue

- Page 2** 2012 Program Chair on Training
- Page 4** Read Good Works
- Page 6** Team Highlight: CERT Australia
- Page 8** EduC Update
- Page 9** Transits II Update
- Page 11** Software development for Incident Response

We look forward to receiving your feedback.
First-educ@first.org



2012 Program Chair on Training

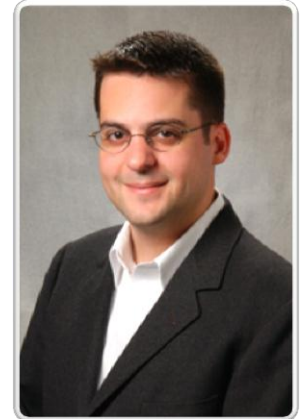
Jose Nazario – How the Program Committee is Injecting Training into the Program, Malta 2012

As the Program Chair for the FIRST 2012 conference in Malta, one of my goals is to increase the value of the conference to attendees in several ways. The conference affords a unique setting for people. First, it's global in scope, allowing teams to meet their peers across the world and build their connection base. Second, it's one of only a few conferences dedicated to the incident response community, a need that is often unmet at other security conferences. Third, its duration - five days - affords some interesting possibilities. To that end, the conference's program committee is seeking to structure an event that will engage the audience and deliver lasting value in the finest of FIRST traditions.

Some of the restrictions we have on what is available to us at the venue in Malta include space. While parallel tracks of hands on training would be ideal, we lack the space needed to run hands on events as a fourth track. Secondly, we have received pushback on the idea of making Monday a hands on, training day (as was done in Seville, Spain, for example). Finally, full teams rarely get to attend the global meeting and instead mostly attend the regional FIRST TC meetings, where training is a major focus. With these kinds of limitations in mind, a focus on education through hands on engagement has been foremost, but we'll attempt them in different ways.

Seeing that approximately half of all conference visitors are first-time attendees, the program committee is working on a "Technical Foundations" track. This track of talks, which has existed as the "newbie track" in previous years, had been dropped in recent conferences by the organizers. After discussions with attendees about what talks and materials are valuable to attendees, I decided this track was a key ingredient to the conference's educational value. These talks are being actively solicited and include introductions to artifact analysis sites on the web that are more common than they were in the past, introductions to the communities that overlap with FIRST, and much more. These talks normally find the submitters thinking, "People have seen this before and it's not the kind of thing that the PC would like to see." Direct solicitation, on the other hand, allows us to bypass this roadblock to submission.

Another element that we are attempting to deliver this year is an open exercise to all attendees with both technical - e.g. "crackme" type challenges - and non-technical - e.g. engagements with other teams - components. This exercise, we hope, would run all week and be achievable in the participants' spare time between talks. We hope to have this exercise announced on Monday morning, run all week, and have winners announced at the end of the event on Friday afternoon, complete with prizes.



by Jose Nazario
Arbor Networks

Continued from page 2

Reintroduce

Technical
foundations track

SIG meetings

Follow along
presentations

Small hands on
demonstrations
in Geek Zone

Introduce

Open Exercise

DRG Challenge

A second challenge run by the Team Cymru sponsored Dragon Research Group is also being put together by Jacomo Piccolini and is more technically focused. Again, an announcement on Monday morning with a reserved spot on Friday afternoon to announce the results, winners, and the answers to the challenge. We hope this will allow for both fun and education throughout the week, and encourage people to mingle and meet each other.

In addition to the exercises, the organizer is hoping to have some presenters have talks with full rooms of participants designed to be "followed along" actively by the audience. While we cannot guarantee power to all attendees or assistants to work the room, we hope that enough will find that listening while actively attempting some of the technical materials will be valuable.

The overall program has a dedicated 2 hour block for the special interest groups (SIGs) including the Vendor SIG, and hopefully the NM SIG and the LE SIG, as well. Birds of a Feather sessions can also be set up in this timeframe, if needed, which will allow for different content to engage the audience. The NM SIG in Vancouver, for example, was well delivered and is the working model here.

Finally, the organizers are exploring using the Geek Zone areas as a hands on small class area for small groups to explore tools and techniques in a smaller setting as was done in Vancouver. Power and space permitting (see above for the space limitations we have in Malta) this should be a worthwhile setting.

The organizers of the 2012 conference in Malta seek to engage the ever growing FIRST community beyond talks in various aspects through non-traditional hands on means, as well as some traditional means. Together this mix of content will deliver a breakthrough event for the conference and yield significant value for those present.

Thank you.

Read Good Works

What is in my book case

by John Kristoff

I can summarize this article in three words. Read good works. From here on out I'm just giving you examples of what I have read, what I tend to read on a regular basis and what I'm not apt to read. With these in mind, I hope to arrive at a subjective, if not imperfect, look at what constitutes "good works" in order to convince you not to necessarily read what I read, but to opt for quality before expediency.

The standard, and still best networking book, is W. Richard Stevens' *TCP/IP Illustrated Volume 1*. If you know this book, let me stop you before you object, citing its last publication date. As of this writing, a second edition was just released in late 2011. Stevens passed away over a decade ago, but thankfully another Internet luminary, Kevin Fall, has been passed the torch and provided the community with a long desired update. Unfortunately I've not yet given the revised edition a thorough reading so I'll reserve final judgment, but if it is nearly half as good as the original it will be worth getting your hands on a copy.

A close second to the Stevens book is Radia Perlman's *Interconnections*. Radia's insight covers not only how things work, but why they are designed the way they are. She compares competing designs and holds no punches when opining the drawbacks of a design that we are often left with in the end. While the current edition is over a decade old and still holds up well, Radia has hinted at the possibility of updating it again. Since Radia is alive and well, if she produces it, it'll be a sure winner.

Now I'll quickly shift gears and annoy a great deal of programmers who have already settled on their language of choice. I'm talking about Perl. I don't consider myself a programmer and in fact my standard line is I don't know Perl, I know **Combat Perl**.



John Kristoff
Team Cymru

Yet, with increasing frequency every year I find myself writing more and more of my own tools. Since I started with Perl many years ago, I've just kept on using it. At this stage in my Perl battles, my absolute favorite book is *Damien Conway's Perl Best Practices*. This book has greatly helped me write not only nicer looking Perl code, but overall, better quality scripts. He lays down a number of helpful considerations in code layout and usage that all Perl battlers ought to heed.



Continued from page 4

If I'm not using Perl I'm probably just writing a shell script. The most frequent book I reach for here is a bit of an old, obscure one. I'm talking about *Bruce Blinn's Portable Shell Programming*. It's so old that if you find it, you'll see that it comes with a 3 1/2" floppy disk. Nevertheless, it is concise and provides all the helpful shell syntax I need to write scripts no matter what UNIX OS I happen to be on.

If I'm not using Perl or shell scripts, I should probably not be writing code, but alas no one has taken away my compiler yet and I do find myself dabbling in C from time to time. The only book I really rely on for those projects is *K&R's The C Programming Language book*. No serious geek should be without a copy.

Often times I'll get some of my best ideas by reading refereed research papers or edited journal articles. There are a handful of organizations that frequently put out high-quality papers. Three of the best are *USENIX, ACM and IEEE*. There are a handful of folks where if their name is on the paper, it is more likely worth reading than not. This includes *Nick Feamster, Vern Paxson, Jennifer Rexford and Stefan Savage*. I also keep an eye on *IETF Internet-Drafts* and *RFCs* as they become available, even if there is a bit more noise to wade through there.

Outside of my specific area of expertise, I try to immerse myself in quality books and periodicals. I tend to read about a dozen to twenty full length books a year. Rarely these are books on the best seller list. Instead I opt for books that have wide acclaim, such as having won a Pulitzer Prize or from a Nobel winning author. I lean towards non-fiction, but I do fit in a few fiction books every year. I have a stream of favorite periodicals that cover global affairs, science and literature from an English-speaking perspective. These include *American Scientist, The Atlantic, Boston Review, The Economist, Foreign Affairs, Harper's Magazine, The New York Times and The Wilson Quarterly*.

I may occasionally flip through a handful of podcasts, online lectures, audio books or videos when I want something with a glow in front of my face or that makes noise in my ears. This is a much bigger mix, but two good, lesser known sources of quality, non-fiction material can be found from *The Teaching Company* (aka The Great Courses) and *The Modern Scholar*.

By way of a parting thought I want to be sure you noticed what is missing. I mentioned no blogs, no RSS feeds, no email lists, no conferences and no social networking links. Engaging with the community is hugely important, but less important is what and where. Just showing up to a well-attended conference once in a while is vastly more important than which one it is. Likewise, nuggets of important information show up on key mailing lists, blogs and web sites in a timely manner, but we generally need to spend less time on these interrupt-driver activities than more. They can be important, but they are far less important than we all make them out to be.

Now ask yourself why you're reading this and not something good?

Highlight of an Existing Member

CERT Australia



Australian Government
Attorney-General's Department



Who is CERT Australia?

CERT Australia is the initial point of contact for cyber security incidents impacting upon Australian networks.

Our primary responsibility is to work with the private sector in identifying critical infrastructure and systems that are important to Australia's national interest, based on an assessment of risk, and to provide these organisations with information and assistance to help them protect their information and communication technology infrastructure from cyber threats and vulnerabilities. This is also achieved through trusted information exchanges between the Australian Government and Australian businesses on cyber security issues. In working to protect critical infrastructure including banking, water, energy generation, transportation and telecommunications, we play a part in ensuring that those services that all Australians rely on are secure and resilient.

CERT Australia is also a source of cyber security information for the Australian community and the point of contact for Australia's international cyber security counterparts. We have a coordination role with the Australian business sector in the event of a serious cyber incident, and we provide Australians with information on cyber threats so that they can better protect themselves.

How were we formed?

Australia, like many other nations, has an ever increasing reliance on information and communication technology in all aspects of life. However, we also face an increasingly sophisticated and hostile online security environment and emerging threats that do not respect traditional jurisdictional boundaries.

The Australian Government E-Security Review 2008 found that Australia's Computer Emergency Response Team arrangements would benefit from greater coordination in order to better respond to these challenges. As such, it recommended the formation of a new Australian Government national CERT. In November 2009, the Attorney-General announced that the new national CERT would be called CERT Australia.

We were formed therefore to provide an initial point of contact for cyber security information for Australia and to coordinate Australia's cyber event response arrangements, nationally and internationally.

In 2009 the Australian Government also published its **Cyber Security Strategy**¹. This strategy defines the most critical Australian businesses as **Systems of National Interest (SNI)**; those which, if rendered unavailable or otherwise compromised, could cause significant harm to Australia's economic prosperity, international competitiveness, public safety, social wellbeing or national defence and security.

Continued from page 6

Our engagement with business centres on those organisations that provide such services, and provides access to information not otherwise available, to support effective risk management.

What tools and assistance does CERT Australia provide?

CERT Australia provides access to specific security training for Australian security practitioners from SNI's; such as the Advanced SCADA **Security Red/Blue Team** training² at Idaho National Laboratories' in recognising and responding to cyber attacks on Supervisory Control and Data Acquisition (SCADA) systems.

Where appropriate, we also provide direct technical assistance in response to a cyber intrusion for Australian businesses.

CERT Australia encourages all Australian businesses to consider, and where appropriate, adopt recommended strategies from the list of the Top 35 strategies for mitigating targeted cyber intrusions³ released by the Australian Defence Signals Directorate (DSD). The **top four of these recommendations** are assessed to have mitigated over 80% of intrusions responded to by DSD.

Although we don't currently offer CERT-related training programs, CERT Australia has assisted with other programs hosted by international CERT teams. We welcome contact from other CERT teams around the world in the interests of fostering technical information-sharing and cooperative incident response operations.

How to contact CERT Australia

You can contact CERT Australia via email: info@cert.gov.au

Or phone us on: +61 2 6141-2999

CERT Australia personnel answer 9:00 am - 5:00 pm Australian Eastern Standard Time (GMT +10) on working days and use a 24/7 watch office for out-of-hours contact. CERT Australia staff are on-call for emergencies during out-of-office hours and on weekends and public holidays.

Our address is: CERT Australia
Attorney-General's Department
3-5 National Circuit
Barton, ACT, 2600
AUSTRALIA

We also have an office in Brisbane, Queensland

¹ [http://www.ag.gov.au/www/agd/agd.nsf/Page/CyberSecurity_CyberSecurity?open&query=cyber security strategy](http://www.ag.gov.au/www/agd/agd.nsf/Page/CyberSecurity_CyberSecurity?open&query=cyber%20security%20strategy)

² http://www.inl.gov/scada/training/advanced_scada.shtml

³ <http://www.dsd.gov.au/infosec/top35mitigationstrategies.htm>

EduC Update

CVSS 2.0 Training

FIRST-Trainers

Model CSIRT?

CVSS 2.0 Training

The EduC is working with the CVSS SIG to augment their existing on-line resources. We will be reorganizing some of the material into a course like structure. In addition, we will also develop a number of interactive exercises. The new material has an estimated delivery date of mid-June. Just in time for the 2012 conference in Malta.

FIRST-Trainers

The EduC is looking to create a community for the trainers in our membership. Our goal is to provide an environment where those who offer and who are conducting training can exchange ideas. We will also work towards creating a web presence for trainers to announce upcoming training and provide information about their courses. Further, this community will receive requests for training received by FIRST.

We are in the process of identifying a community manager.

Model CSIRT?

EduC is looking for 3 types of CSIRTS (small, medium, and large).

- Small: 1 – 6 employees
- Medium 7 – 20 employees
- Large 20+ employees

One of our long term goals is to develop general training plans to assist CSIRT members. If your team is interested to be a test subject, we would like to schedule time during our Annual Meeting to work with you. You would get an interesting view from EduC experts in training and we would be able to gather and generalize your requirements. Let us know if you are interested!!!

Contact us at FIRST-EduC@FIRST.org



EduC Annual Meeting

Room: TBD

16th June 2012
09:00 to 16:30
Working Group

17th June 2012
09:00 to 14:00
Working Group
14:00 to 17:00
Train the Trainer

The Train the Trainer is a series of presentations combined of Lessons Learned and “Doing what works.” It would be of interest to all members.

TRANSITS I and II

State of Affairs – January 2012

by Don Stikvoort

TRANSITS Coordinator, FIRST Liaison Member



Where do we stand with TRANSITS I and II now in 2012?

First some keywords for your benefit:

TRANSITS I: introductory CSIRT training; interaction, sharing, networking.

TRANSITS II: advanced topics for CSIRTS; emphasis on technical but including communication skills; flexible to allow addition of other advanced topics.

Both courses are available for the world at a not-for-profit base. In fact - "not-for-profit" is the hallmark of TRANSITS. As we all think that TRANSITS is a worthwhile contribution to the global effectiveness of incident response.

Baseline: with TRANSITS you get top notch experts to train your staff at very reasonable prices!

TRANSITS no doubt deserves the label of most popular introductory course for CSIRT members worldwide. It has been around for over 10 years now and has been given many times, on all continents except Antarctica!

TRANSITS derives from Europe and is very popular there. TERENA, the not-for-profit copyright holder of the materials, organises two courses per year - and we always have more candidates than we can accomodate. But also several other CSIRTs in Europe organise their own TRANSITS courses regularly!

Outside Europe, the course has been especially popular in Japan, Korea and various Latin-American countries. But it has also visited China, the Middle-East, South Africa, Georgia, ...

The course can be run at a relatively low price compared to commercial trainings, making it available to all developing CSIRTs, in all regions.

This is because the materials have been developed by renowned experts in our field at marginal cost. The same not-for-profit principle holds for reimbursing the tutors.

Where are we right now with TRANSITS ?

The TRANSITS most of you know has been re-labeled TRANSITS I. It is a course that basically runs over two to three days rich in content and fun.

There are four basic modules. Organisational, legal, operational and technical. The operational and technical modules have been completely rewritten - the new operational module has already been field

Continued from page 9

tested in autumn 2011 - "technical" new style will be launched in March 2012.

"Organisational" will be revised next summer - to be followed by "legal".

In fact, from now on, every module is planned to be thoroughly revised every 2-3 years. This makes clear that TRANSITS may be around for a long time but keeps going strong!

Together these modules provide an excellent introduction to CSIRT work.

Very useful for new or emerging teams - and for new team members of existing teams. The way we do TRANSITS is to allow much interaction - and we include a roleplay exercise based on the body of ENISA CSIRT Exercises.

The next European TRANSITS I course is in Porto (Portugal) from 28-30 March. See

<http://www.terena.org/activities/csirt-training/transits-i/porto/> .

Subscribe now!

So what about TRANSITS II? This is for advanced topics. The first official TRANSITS II was done in April 2011 in Zürich. It was a great success. We take three full days for TRANSITS II. Over a day of forensics, one day of netflow and then a short training in human communication aspects, plus a more elaborate CSIRT exercise based on the ENISA Exercises.

The next TRANSITS II is due 2-4 April in Prague. See <http://www.terena.org/activities/csirt-training/transits-ii/prague/> . You can still subscribe!

And what if you want to organise your own TRANSITS I or II course? That is possible. The license fee for either has been set at € 600 - this is used to maintain and improve the course materials. More information see <http://www.terena.org/activities/csirt-training/conditions.html> - or get in touch with Jim Buddin, Kevin Meynell or Don Stikvoort in Rome or via e-mail.

So as you can see TRANSITS is alive and kicking. The courses have benefited many teams worldwide over the past ten years, and we are going to continue on that course in the years ahead. This is possible due to the contributions of many of you - for which we heartily thank you here!



Incident Response Software Dev.

Four Ways to Improve Your IR Software Development

by Chris Horsley

Founder, CSIRT Foundry

<http://csirtfoundry.com/blog/>



If you run an incident response(IR) team, chances are there's a shadow team lurking in your midst: a software development house. Developing one-off and specialised tools is an inevitable part of IR. These can range from small scripts to full-featured web applications. IR teams may not even have a formal software development team; instead, incident responders use whatever time they have between tickets to develop tools.

Although not specialising in software development, these same IR teams can recognize improvements through a few simple and inexpensive development practices. Below, we outline four quick wins that could increase full-time and part-time developers' effectiveness.

1. Give 'em a break from the ops room

IR and software development are two separate mindsets. Incident response is about waiting for things to happen, and then reacting. Project work, such as software development, requires long stretches of intense amounts of concentration. Although developers need to understand the IR team's processes and pain points, not giving your developers a break from the bustle of IR invites problems.

An important factor for programmers is called "flow". Flow is the highly productive state of being fully immersed in a mental activity, like developing software. Programmers keep the state of what they're doing in a mental mind map: the order functions call each other, what each variable contains, and how data flows through the system. Phone calls, e-mail alerts, or chirping text messages destroy "flow", and drags programmers from the mental world they've been building back into the physical world.

It's a disconcerting experience. Once the developer gets back to their development work, it will take a good amount of time to get back into a state of flow again. Interrupt them enough, and they'll soon become adverse to even getting into a deep state of concentration, for fear they're going to be shaken out of it. Procrastination sets in. This is not just the case for incident interruptions; multiple unscheduled meetings or drop-ins to their desk can have the same effect.

So, how do we fix this? One strategy might be to offer anyone doing development a quiet environment away from the ops room. Failing that, a pair of headphones and a "do not disturb" policy can suffice. Those working on project roles are sheltered from phones and conversations, and can get into that all-important flow state. If you need to continually rotate staff back into the operational pool, try to let those staff know how long they have allotted for project work - a day, a week, or a month. It's impressive what can be done in a limited time frame, as long as that time frame is known.

Continued from page 11

2. Use Source Control

As well as making sure your developers are working in a conducive environment, getting good practices in place are also important. Often in the case of having only one or two developers, any overhead other than writing code is regarded as overkill. Documentation is a luxury, and testing is something that happens after going live.

One day, though, your heroic solo developer will move on, or if you're lucky, additional developers will come on board. At that point, the laissez-faire attitude that worked fine until now starts to fall apart.

When people move on, or your lonely developer gets some help, a systematic way of managing and recording changes to your software is essential. One key way to do this is via a source control system, where code changes are checked into a repository and tracked. Previous versions of files can be recovered, multiple developers can work on the same code base without disturbing each other, and unstable new features in development can be easily segregated into quarantined areas which don't interfere with stable production code.

By performing regular commits, not only do you get a full, incremental revision history, but also comments detailing what each change achieves. When you have to track exactly when a critical bug was introduced six months after the fact, the ability to move back through a commented set of revisions is invaluable.

So, which source control system should we use? That's a religious debate if there ever was one, but one of the strongest contenders at the moment would be Git. Git has achieved staggering uptake in the last few years, and as a former Subversion user, it really is streets ahead in the way it handles merging while having a lower overhead. Setting up a new local repository is as easy as running:

```
$ git init
$ git add myscript.py
$ git commit -a -m "My first commit"
```

Git is a distributed source control system, meaning that scripts modified away from the office can have their own local changes tracked too, to be merged into a master repository later.

For refugees from centralised source control systems such as Subversion, there are a few new concepts to get around, but plenty of tutorials to help out.

3. Perform automated testing

The tedium of testing code by manually inputting various values is something most developers spurn in favour of writing new code. However, what if your developers could write code to test code?

Automated testing allows developers to write test cases which send inputs to software, and test that outputs work as expected.

Continued from page 12

A common problem in software development is introducing a new feature which breaks an old feature. Using code-driven testing, the entire historical body of tests can be run as one, ensuring that not only new features work correctly, but old ones too. There's something distinctly satisfying about adding a new piece of code and seeing all tests clear.

Some types of software are easier to test than others. For example, compared to command line tools, tests for web sites with complex user interface interactions can be difficult, but not impossible, to automate. Independently testing the functions of any libraries used by a piece of software is also a helpful technique.

Which testing framework should you choose? It really depends on your language, budget, and environment, but a visit to Wikipedia's unit testing framework listing would be a great start. (http://en.wikipedia.org/wiki/List_of_unit_testing_frameworks)

4. Use project management tools

The term "project management" is enough to make some solo developers sigh at the prospects of increased overheads, but project management is something that really can be done incrementally with little added time costs.

The first essential for any project is bug tracking. When problems are found in the software, whether they're critical security problems or "one day" feature requests, we need a place to record them. The second essential feature we need is documentation repository for the software: perhaps a roadmap or user manual.

A common temptation is to reach for Excel and Word, but we can do a lot better than document files buried on a file server somewhere. There's a vast variety of software project management tools out there, but two great starting points are Github and Redmine.

Both are web applications that allow multiple developers and users to lodge issue tickets, add documentation to a wiki, see changes to the software repository, and a lot more besides. The chief difference between them is that Github is nominally a hosted application, where Redmine is self-hosted. Github is the natural choice for projects available to the public or shared between organisations, while Redmine is a good choice for those projects you're not quite comfortable hosting in someone else's datacenter.

Summary

Whether your team's software development is handed by dedicated staff or by incident handlers in-between incidents, any of these steps will help improve the state of your software tool development:

1. Let your developers have a break from the ops room
2. Use source control such as Git
3. Perform automated testing
4. Use project management software such as Github and Redmine

As well as a boost in productivity, you're setting the base for an easier transition when you finally make that next hire for a full time developer.