Section A – Personal Data:

a. **Name**: Stefan Esser

b. **Email Address:** stefan@antid0te.com

c. **Company:** Antid0te UG

d. Brief biography:

Stefan Esser is best known in the security community as the PHP security guy. Since he became a PHP core developer in 2002 he devoted a lot of time to PHP and PHP application vulnerability research. However in his early days he released lots of advisories about vulnerabilities in software like CVS, Samba, OpenBSD or Internet Explorer. In 2003 he was the first to boot Linux directly from the hard disk of an unmodified XBOX through a buffer overflow in the XBOX font loader. In 2004 he founded the Hardened-PHP Project to develop a more secure version of PHP, known as Hardened-PHP, which evolved into the Suhosin PHP Security System in 2006. Since 2007 he works as head of research and development for the German web application company SektionEins GmbH that he co-founded.

In 2010 and 2011 he got a lot of attention for presenting about iPhone security topics and supplying the jailbreaking scene with an exploit that survived multiple updates by Apple.

Section B – Training Class Data

a. Title of Training Class:

MacOS Kernel Internals for Malware Researchers and Endpoint Protection Vendors

b. Brief Description of Training Class:

For several years now we have offered MacOS and iOS kernel exploitation and kernel internals courses that were designed from the point of view of vulnerability/security researchers and exploit developers. We have now added this new variant to our training course that focuses on kernel (security) internals from the point of view of MacOS malware researchers or endpoint protection vendors.

Unlike previous MacOS kernel internals courses this course will focus more on MacOS rootkits/malware and freely available endpoint protection software. Trainees will learn about the kernel internals involved in the internal workings of MacOS malware and those used by protection software. During the training trainees will work with MacOS installations inside VMs and in multiple hands-on tasks learn about the internals of MacOS malware and a selection of free available protection software.

ATTENTION: This course is not meant as a guide how to write MacOS kernel malware. Instead the focus is on detection and defense.

Topics:

Introduction
————————

* How to set up your Mac for Kernel Malware Research
* How to write your own kernel extension

Low Level x86_64
———————————-

* Differences between x86_64
* Exception Handling
* Hardware Page Tables
* Special Registers used by MacOS
* …

Kernel Debugging
————————————

* Panic Dumps
* Using KDP on MacOS
* Advanced Usage of KDP on MacOS (python scripting)

Kernel Internals
———————————

* Structure of the Kernel Source Code
* Implementation of Mitigations
* Mach messages and IPC
* Security: MAC Policy Hooks, Sandbox, Code Signing, Kauth, socket filter
* Filesystems, networking stack
* …

Kernel Drivers/Extensions
——————————————————

* Writing your own kernel driver
* Limitations for non-Apple drivers and how to circumvent them
* Kernel driver loading and code signing

Kernel Malware
————————————————

* Hooks and how to detect them

* (Un)Hidding Files

* (Un)Hidding Network communication

* detecting bypasses

* ...

Kernel Memory Forensics
————————————————

* MacOS memory acquisition

* MacOS memory forensics with volatility

Persistence
——————--

* How does MacOS malware persist?

* What are the defenses?

c. Pre-requisite of Training Class:

i. Student:

* Students must be capable of understanding/programming code in C/python

* Students will get an introduction to low level CPU security features (x86_64) as part of the course

ii. Hardware:

* Macbook capable of running latest OS X/MacOS in a VM

iii. Software.

* IDA Pro

* OS X El Capitan/Mac OS

* Xcode with latest SDK

* VMWare Fusion with Mac OS VM