セキュリティ・キャンプ全国大会 2014

セキュアなシステムを作ろうクラスの説明

【 目 次 】

概要

　このクラスは何らかのシステムの開発を通じて、セキュリティについて学ぼうというクラスです。2012 年までは OS に関する開発に限定していましたが、2013 年からは OS に限定せずに少し幅を広げています。

　システムの開発にはたくさんの分野あり、それぞれによって開発手法や指導方法も変えたほうが効果的だろうという考えに基づき、クラスを複数のゼミに分割しています。どのゼミでセキュリティについて勉強したいかを選んでください。

　選べるゼミは一つだけで、第二希望などは取りません。その代わりよい学生が多く集まったゼミは担当講師を増やしてたくさんの学生をとってもらいますので、迷わず希望するゼミに応募してください。

　一部、同じようなテーマが複数のゼミにあります。そういうテーマの場合に、どう選べばいいのか迷うかもしれませんが、そのときはそのゼミが他にどんなテーマで募集しているかを見てください。そうすればどんな学生が集まるゼミなのかを想像できるはずです。

　ゼミごとに応募用紙が用意されていますので、間違えないように注意してください。

　このクラスには次の 5 つのゼミがあります。以下にそのゼミの内容の説明を書きます。

・ゼミ名「OSECPU-VM ゼミ」（ゼミ長：川合秀実）
・ゼミ名「組込みのセキュリティを考えるゼミ」（ゼミ長：坂井弘亮）
・ゼミ名「ＯＳの見える化を考えるゼミ」（ゼミ長：半田哲夫）
・ゼミ名「システムソフトウェアゼミ」（ゼミ長：忠鉢洋輔）
・ゼミ名「ルーター自作ゼミ」（ゼミ長：竹迫良範）

（1） ゼミ名「OSECPU-VM ゼミ」（ゼミ長：川合秀実）

■ゼミの概要

セキュリティを意識した VM・OS・言語処理系を設計・開発してみようというゼミです。
この作業を通じてセキュリティについて学びます。

このゼミでは、どのテーマを選ぶ場合でも、自宅に Windows もしくは Linux の PC（MacOS でも可）があって、それを使って自力でプログラムを作れることが必要です。

このゼミではテーマは一つしかないのでテーマ選択はありません。

OSECPU-VM というのは、2013 年のセキュリティキャンプで作ったセキュアなシステムです。今年もこれを学生のみなさんと一緒に作りたいと思います。OSECPU-VM について詳しいことは http://osecpu.osask.jp/wiki/ を見てください。

一緒に作るとは言っても、何か役割分担で作業を押し付けられるというわけではなく、
基本的には各自の好きなことをマイペースでやってもらおうと思っています。コアとなる
処理系の開発を手伝ってくれるのはもちろん歓迎ですが、OSECPU-VM 用のアプリを作って
遊んだり、ドキュメントの不備を見つけて書き足してもらったり、OSECPU-VM を移植したり、
独自拡張してみたり、OSECPU-VM 上で動くシェルを作ったり、セキュリティーホール探しを
したり、できることはたくさんあります！・・・どんなことを通じてでも、セキュリティに
ついて十分に学ぶことができます。

Q: OSECPU-VM に関係のない開発がしたいのですが・・・
A: 申し訳ありませんが、昨年とは異なり今年は OSECPU-VM 関係のみのテーマに一本化して
います。これは単に指導を効率よくするためです。これはあなたの開発テーマを否定して
いるわけではありません。むしろ既に自分の開発テーマを持っているあなたにこそ
OSECPU-VM の開発の経験を数か月ほど体験してみることをお勧めします。きっと多くを
学べるでしょう。そして学んだことを今後の開発に役立てることができるでしょう。
セキュリティに対する考え方が変わるかもしれません。

Q: 興味があるのですが実力が足りるかどうか心配です。
A: 応募者多数の場合は、もちろんやる気と実力のありそうな人から選んでいく予定ですが、
応募者が少なければ実力がそんなになくても選ばれると思います。応募者が多いか少ないかは
年度によってまちまちなので、何とも言えません。
でも、そうやって応募状況におびえるようでは、まだまだやる気が足りないと思いませんか？
そんなの気にしないで全力で応募してしまえばいいんです。そしてそういう人が結局は

選ばれるのです。
実力や希望に応じてやることを相談して決めますので、選ばれてしまったけどどうしよう、
なんていう心配は無用です！

Q: 応募前から開発コミュニティに参加しておくことは有利ですか？
A: もちろん有利です。それはやる気があるという何よりの証拠です。開発コミュニティ内では
適当なハンドル名を名乗ることになると思いますが、応募用紙にそのハンドル名を書いて
もらえたら、選考時に大いに参考にします。あなたのスキルが分かりやすくなります。
ただし、参考までに申し上げておきますが、2013 年の当ゼミの参加者は、3 人全員が応募時は
開発コミュニティに参加していませんでした。ですからこれをそれほど深刻に考える必要は
ありません。
また開発コミュニティに参加していれば、セキュリティキャンプに落選してしまっても、
ある程度の指導を受けることができます。それはそれで悪くないことだと思いませんか？

Q: 実はセキュリティにあまり興味がないのですが・・・
A: 笑。正直者ですね。いいですよ、そんなあなたも歓迎です。私の役目はセキュリティに
関係のない作業から関わってもらって、だけどそのうちにあなたにセキュリティの重要性を
分かってもらって、その上でどうすればセキュアなシステムが作れるのかを考えてもらえる
ようにすることです。応募時点でセキュリティに興味がないからといって門前払いにする
ことはありませんので、どうぞご安心ください。そもそも私だってセキュリティキャンプに
関わる前は、セキュリティにあまり興味がなかったくらいです。

Q: C 言語もアセンブラも経験がゼロですが大丈夫でしょうか？
A: まったく問題ありません。他のプログラミング言語の経験があれば十分です。

（2）ゼミ名「組込みのセキュリティを考えるゼミ」（ゼミ長：坂井弘亮）

■ 概要

本ゼミではマイコンボードなどの「組込みシステムでのセキュリティ」を扱います．また本年度よりテーマの幅を広げ，CPU などの「コンピュータ・アーキテクチャ」を主眼に置いたセキュリティについても考えます．

よって「組込み関連のテーマ」と「アーキテクチャ関連のテーマ」があります．
組込み関連のテーマは，とにかくマイコンボードをいじりたおす感じです．
アーキテクチャ関連のテーマは，様々なコードを書いてエミュレータなどでいろいろ試す感じです．アセンブラも多く扱います．

組込みシステムで考えられるセキュリティ対策はＰＣとは異なった，独自性の強いものになります．組込みシステムではどのようなセキュリティが考えられるのか，実際にソフトウェア開発を行うことで，ものづくりを通して考えてみましょう．

またセキュリティの本質を理解するためには，コンピュータ・アーキテクチャを知ることがとても大切です．アーキテクチャを知ることで，コンピュータの動作原理から見直した根本的なセキュリティ対策をとることができます．

本ゼミでは各自で開発テーマを決めて，セキュリティに関する組込みやアーキテクチャ関連のソフトウェア開発を行います．キャンプ前から当日にかけてオンラインで相談をしながら開発を進め，キャンプ中になんらかの成果物を出してデモができるようにすることを目標とします．

なんだか文章が固くて難しそうですが，オンラインでの事前学習からじっくり時間をかけて進めますので大丈夫ですし，組込みにせよアーキテクチャにせよ，きっとすごく面白いし勉強になります．

開発テーマは選択テーマの中から選ぶか，持ち込みテーマを提示するかになります．
持ち込みテーマの場合には，自身でテーマ内容を提案することになります．

ＰＣとは異なり，組込みは多種多様・住み分けの世界です．このためゼミの内容に多様性を持たせたいと考えて，選択テーマには多数の候補を上げてあります．しかし，すべてのテーマに一人ずつを割り当てるわけではありません．ただしやりたいことが複数あれば，複数のテーマを組み合わせて選択しても構いません．テーマを自分なりにアレンジしても構いません．

また応募状況によって，同一のテーマに２人以上を割り当てることもあり得ます．さらに実施テーマは完全に固定にするわけではなく流動的です．例えばあるテーマの次段階として別テーマに移るなど，進度や興味の変化に応じて柔軟に，臨機応変に進めます．

また設計や実現の方法については講師側から提案もしますが，自身のアイディアがあれば歓迎します．ぜひ積極的に提案してください．

■ 応募条件

本ゼミの応募条件は以下のようなものです．条件を満たしていれば，組込みプログラミングや低レイヤーやセキュリティの知識・経験は問いません．

・モノづくりが好きで好きでしょうがなく，昼も夜もモノづくりのことばかり考えている，というかた．

・なんらかのソフトウェア開発の経験があること．(組込み分野でなくても構いません)

・アセンブラなどの低レイヤーの学習に抵抗感が無いこと．事前学習で勉強するので，始めから知っている必要はありませんが，学習することに抵抗感が無いこと．
アセンブラやコンピュータ・アーキテクチャなどの低レイヤーの学習を「面白そう」と思えること．

・キャンプ当日まで，自宅で事前学習ができること．
キャンプ期間中だけですべてを開発することは難しいのと，短いキャンプ期間を有意義に使いたいため，キャンプ当日までにオンライン指導のもとで事前学習をしていただくことが前提になります．ただ，学校の試験期間や家族旅行などで空き期間があっても構いません．(配慮します)

・開発物の動作検証には実機もしくはエミュレータなどの実行環境を想定しています．
エミュレータでも可ですが，実機で行うことを想定している場合には，自身でハードウェアや開発環境などの準備ができて当日持参できること．
例えば普段から使っているマイコンボードがあるのでそれを題材にして，当日もそれを持参して使う，などです(エミュレータ利用の場合は，その限りではありません)．
例として，昨年は Arduino Uno を題材にして持参した参加者がいました．

■ 各テーマに共通の事項

開発ターゲットや環境については，以下の条件を満たしていればとくに問いません．

参加決定後に，相談の上で決めましょう．

・既存の組込みＯＳやシミュレータなどをターゲットにして開発を行う場合には，オープンソースのものであればあとは問いません．組込みＯＳにはゼミ長の坂井が開発している「KOZOS」を推奨しますが，他でも構いません．

・対象アーキテクチャは問いません．

・開発環境は，誰でも自由に利用できるものならば問いません．一応 gcc を推奨します．

■ 選択テーマ一覧

・テーマ#01「マイコンボードでの不正実行の検出機能を作ってみよう」(組込み)

マイコンボード上で脆弱性のあるソフトウェアが外部から攻撃された場合を想定して，アプリケーション・プログラムの挙動を監視し，不正実行を判断して中断・ログ保存するようなセキュリティ機能をマイコンボードに実装します．

ログはマイコンボード上の制限されたメモリ内に保存することと，ＰＣ側にアップロードできることが必要になります．

・テーマ#02「セキュリティソフトを作ってみよう」(アーキテクチャ)

UNIX 系のシステムでは ptrace()のようなデバッガ向け機能を利用することで，アプリケーションの挙動を高度に制御することができます．

デバッガ向け機能を利用してアプリケーションの挙動を能動的に監視しチェックすることで，ＰＣ上で動作するセキュリティソフト「もどき」を作成してみましょう．

・テーマ#03「アプリケーションの挙動を見える化してみよう」(アーキテクチャ)

「セキュリティソフトを作ってみよう」の発展系です．

ptrace()のようなデバッグ機能によりアプリケーションの挙動を見張り，レジスタやスタックの状態を動的に取得し変化をリアルタイム表示することで，動作を見える化してみましょう．さらにバッファオーバーランの脆弱性の攻撃等を実験的に行い，視覚的に見てみましょう．

なお本テーマはＰＣでの開発を想定していますが，マイコンボードをターゲットとした

開発でも構いません．(この場合は GDB によるリモートデバッグ機能が利用できます)

・テーマ#04「セキュアな組込みＯＳを自作してみよう」(組込み)

既存の組込みＯＳに独自に機能を追加することで，セキュアな組込みＯＳを自由なアイディアのもとで開発してみます．

何らかのオープンソースな組込みＯＳが動く，マイコンボードをターゲットとします．

・テーマ#05「アーキテクチャの違いを検証してみよう」(アーキテクチャ)

バッファオーバーラン等による脆弱性の既存の攻撃方法を，様々なアーキテクチャについて検証し横断的に比較することで，スタックやレジスタの扱い，命令コードの特徴，命令セットの違いなどによるアーキテクチャごとの脆弱性の特徴を検討します．シェルコード挿入や ROP，Exploit コード作成をエミュレータ環境により実験します．

想定するアーキテクチャは x86，ARM，MIPS，PowerPC，SH，H8，V850，AVR，RX，6502，Z80，VAX などですが，可能な限り多く扱い横並びにして比較することを目的とします．

・テーマ#06「エミュレータでの不正実行の検出機能を作ってみよう」(組込み)

エミュレータ上で組込みＯＳのカーネルやアプリケーション・プログラムを動作させ，ＯＳへのシステムコールや特権命令の実行，メモリ読み書きなどの詳細なトレースを採取する機能をエミュレータに実装します．さらに実行中のプログラムの動作のチェックを行うことを可能にします．

エミュレータでの開発がベースとなるため，マイコンボードは利用しません．

・テーマ#07「組込みシステムをセーフティに設計してみよう」(組込み)

組込みシステムでは「セキュリティ」(脅威対策)に対して「セーフティ」(機能安全)という考え方も重要になります．

エレベータ制御や踏切制御等を題材として制御プログラムを作成し，シミュレーション動作させ，ドアの開閉・ボタンの押下・人がドアにはさまれたことの検知などをして適切に振舞うかどうか，検証してみましょう．なおシミュレータも自作することを前提とします．シミュレータでの確認がベースとなるため，マイコンボードは利用しません．

・テーマ#08「プロセッサのセキュリティアシスト機能を設計してみよう」(組込み)

プログラムの不正実行や不正領域アクセスなどの検出を考えた場合，既存のプロセッサに対して専用のハードウェアアシスト機能を搭載しＯＳ側から適切に利用することで，高速かつ確実に検出できる可能性があります．

既存のプロセッサに追加する形でそのような専用機能や特殊命令を設計し，ＯＳと協調してセキュリティを確保する方法を検討します．さらにプロセッサのエミュレータ上で特殊命令の実装を行い，命令の動作やＯＳからの利用などの動作検証を行ってみます．

エミュレータでの開発がベースとなるため，マイコンボードは利用しません．

・テーマ#09「セキュアなインターフェースを設計してみよう」(アーキテクチャ)

既存のアーキテクチャでも ABI(Application Binary Interface)を工夫することで既出の攻撃手法を無意味にしセキュリティを高めることができる可能性があります．

セキュアな ABI を定義し，アセンブラベースでプログラミングすることで効果を確認します．可能ならばコンパイラやリンカを改造し，そのような ABI でコード生成してみます．

アーキテクチャは一つに絞らず，x86，ARM，MIPS，SH といった様々なものを扱います．

・テーマ#10「低レイヤー学習のための教材を作ってみよう」(アーキテクチャ)

セキュリティ理解のための低レイヤー学習の促進を目的として，2013 年には「アセンブラ短歌」や「バイナリかるた」「アセンブラ・クロスワード」といった新しいコンテストや競技，ゲームが考案され，実施されています．

勉強会で実施できるような題材として，そのような新しい教材を考案し，開発してみましょう．アイディア出しだけではなく，自身で実装し普及させることを前提とします．

（3）ゼミ名「ＯＳの見える化を考えるゼミ」（ゼミ長：半田哲夫）

■概要

セキュリティ侵害の脅威にさらされる領域が拡大し、実際に侵害される事例も多発
しているかと思います。そして、領域が拡大し事例が多発する理由の一つとして、
システムをブラックボックスのまま利用していることが挙げられるかと思います。
システムを安全かつ有効に活用するためには、何でも専門家や有識者に丸投げすれば
よいという姿勢ではなく、自分で内容を理解しようとする姿勢が必要だと考えます。

このゼミを担当する熊猫さくらは、 Linux カーネル向けアクセス制御機構の開発を
通じて、ＯＳレイヤで行われていることを理解してもらうための試行錯誤を続けて
きました。現在は Linux システムのトラブルシューティングを業務として行っており、
システム障害やセキュリティ侵害を予防するために、まずはＯＳレイヤで何が
行われているのかを理解してもらうことが必要であると考えています。そこで、
熊猫ゼミでは、ＯＳレイヤで行われていることを少しでも見えるようにする方法を
一緒に考えていきたいと思います。

■テーマ

・テーマ#01「 Linux システムの運用監視プログラムを作ろう」

運用監視プログラムは商用製品も含めていろいろありますが、動作が重かったり、
痒い所に手が届かなかったり、エラー処理やデバッグ情報出力を疎かにしている故の
問題に悩まされたりすることも多いかと思います。

そこで、このテーマでは、エラー処理をしっかり行いながらもサクサクと動作する、
特定の目的に特化した、丁寧なプログラムを作ることを目指します。

・テーマ#02「 Linux に関するトラブルへの対処手順と予防策を考えよう」

ＯＳレイヤでのシステム障害（カーネルパニックやハングアップなど）は、事前に
どれだけの備えを行っていたかが、原因究明と再発防止ができるかどうかの大きな
分かれ目になります。

そこで、このテーマでは、 Linux の内部動作について知り、どのような備えが
できるかを考えます。発表してみんなと共有するところまでできるとベストです。

・テーマ#03 持ち込みテーマ

基本的に Linux カーネルまたは Linux 向けのシステムプログラミングが関係する
領域でないと対応できませんが、希望内容を見て判断します。

■応募条件

熊猫ゼミに応募する人には、疑問に思ったら遠慮なく質問できる図太さと、自分から
考えて手を動かせる積極性が望まれます。損得勘定抜きで行動できる熱意の持ち主や、
何年でも飽きずに諦めずに続けられる職人気質の持ち主だとベストです。
テーマ毎に望まれる条件としては、以下のものがあります。

・テーマ#01

・既存の運用監視プログラムに対して不満な点や改善したい点を持っていること。

・C言語を用いてソースコード規模が数百行程度になるプログラムを自分で設計して
作成できること。

・テーマ#02

・ Linux システムの内部動作に興味を持っていること、あるいは、
Linux システムの構築／運用／保守のレベルアップをしたいと思っていること。

・テーマ#03

・コーディングを伴う内容の場合には、自分で設計して作成できるスキルが
あること。

（4）ゼミ名「システムソフトウェアゼミ」（ゼミ長：忠鉢洋輔）

■ゼミの概要

システムソフトウェアというと、あまり皆さんに馴染みがないかもしれません。ですが実際、皆さんが普段使っている PC やスマートフォンのアプリケーションは、システムソフトウェアの深い層（レイヤー）に支えられて動いています。アプリケーションを作る人は世界にたくさんいますが、このレイヤーは縁の下の力持ち的な存在であり、あまり目立たないこともあって、システムソフトウェアを作ったり、そのセキュリティを考える人はあまり多くありません。

ですが、大変楽しい分野なのでこの楽しさをシェアしたい、また、これがキッカケでこの分野に貢献しうるエンジニアになってくれたらいいなぁ、というのがこのゼミを開講する目的です。

このゼミでは、システムソフトウェアという、いわゆるオペレーティングシステム（OS）のカーネルプログラミングやシステムプログラミング、ハイパーバイザの実装を中心に扱います。各レイヤーでのセキュリティ機能の実装や、マルウェア検知や CTF サーバの監視運用のためのシステム開発などを通じ、OS や VM、そしてそのセキュリティ技術を深掘りするキッカケにしてもらおうと思っています。

セキュリティはあまり関心が無いけど OS カーネルやハイパーバイザを開発したい人、セキュリティに関心があるし低レイヤーな技術に憧れもあるけどちょっと難しそう・・・というような人も、是非応募して欲しいです。実装を中心に進めますが、基本的にやる気とプログラミングについてある程度経験があれば問題ありません。もちろん、元々この低いレイヤーとセキュリティに興味があり、バリバリと実装出来る人がいれば歓迎します。とても歓迎します。

本ゼミでは、基本的にオープンソースソフトウェアをベースに、事前準備期間とキャンプ期間を合わせて、ある程度の規模のセキュリティシステムを実装することを目標とします。取り組む課題については、参加者の選抜後に参加決定者の興味/趣味やスキルを考慮し、テーマの詳細や規模を調整します。最終的には、成果を勉強会で発表したり、コードを公開することを推奨します。

■応募条件

・授業や講義の課題以外で、主体的にプログラミングやシステム開発に取り組んだことがある人。ただし、自分でテーマを設定するような課題や、プログラミングコンテストなどは、主体的なプログラミングやシステム開発と考えてください。

・事前学習期間に連絡を絶やさず、主体的にセキュリティキャンプに取り組める人。

■ 選択テーマ一覧

テーマ 01:「サンドボックス化によるセキュアなプログラミング」

アプリケーションの脆弱性やバグを無くすことは難しく、大きく複雑なアプリケーションにおいてそれは顕著です。なので、アプリケーションに乗っ取られたとしてもその被害を最小限にするための、アプリケーションのサンドボックス化というセキュリティ技術がいくつかのオープンソースソフトウェアで採用され、実装されてます。代表的な例として Google Chrome や OpenSSH などがあります。

このテーマでは、アプリケーションのサンドボックス化についての設計と実装を学びます。具体的には、アプリケーションサンドボックス化を支援する代表的な実装である Mode 2 seccomp を使い、基本的な使い方を学んだ後、参加者のスキルに合わせた規模のアプリケーションをサンドボックス化することにチャレンジしてもらいます。

また、サンドボックス化の技術は OS に依存した機能（IPC やセキュリティ機能）を積極的に使うため、自分の好きな OS についてもっとよく知る機会になるでしょう。　私はどうしてもこの OS でこのテーマをやりたい！という思いがあれば、是非、応募用紙にそれをぶつけてください。

テーマ 02: 軽量 ruby を使ったセキュアなアプリケーション拡張

このテーマではアプリケーションの拡張機能を軽量 Ruby（mruby）を使って実装することにチャレンジしてもらいます。例えば、ネットワーク通信を解析するツールである Wireshark の拡張機能を軽量 Ruby で書けるようにしたり、Linux コンテナ（LXC など）を軽量 ruby で操作するためのライブラリ作成などを考えています。また、軽量 Ruby を OS カーネルに組み込み、パケットフィルタや動的なコンフィグレーションを実現する機能を実装するといった、少し難易度の高い課題も用意しています。

テーマ 03: クラウド向け OS "OSv" のセキュリティ向上

このテーマでは、クラウド向け OS「OSv」でより安全にアプリケーションを実行出来るようにするためセキュリティを強化する機能を実装することにチャレンジしてもらいます。
（OSv は仮想化環境やクラウド上でアプリケーションを実行する事に特化し、一から設計された新しいオープンソース OS です。）

具体的には、以下のような題材から実装する機能と実現方法を考えてもらいます。
・スタックプロテクタの実装（バッファオーバーフローの防止）
・ASLR（プログラムの読み込み位置をランダム化し不正なコード実行を困難化する機能）の実装
・W^X（メモリ領域のアクセス制御を厳しくする事で不正なコード実行を困難化する機能）の実装
・ロードするプログラムを最小限にする事で脆弱性が発生する可能性を減らすため、OS 機能をモジュール化しアプリケーションに応じて必要な機能のみメモリにロードする仕組みの実装
・ファイヤウォール機能の実装

実装したコードは説明文を付けて開発者メーリングリストへ投稿し、取り込んでもらう所までを目標とします。
OS の低レイヤーな知識やセキュリティに関する知識だけでなく、オープンソースコミュニティにおける開発手法を学ぶ良い機会になると思います。

テーマ 04: 僕、私が考えた最強のセキュア OS の開発

SELinux や AppArmor のようなセキュア OS は、Web 検索にかけるとよく無効にされています。セキュリティを高めるための技術のはずなのに、この有り様です。悲しいですね。このテーマは、誰もが有効にしたくなる（？)、”参加者の皆さんが考えた最強のセキュア OS”をデザインし、プロトタイプの実装にチャレンジするテーマです。このテーマは難しいので、セキュリティにかける熱い情熱を持ち、わりと具体的なアイデアがあって、さらにカーネルプログラミングの経験がある人のみを選抜します。

テーマ 05: 仮想化技術を用いた、ストレージベース侵入検知システムの開発

OS カーネルがマルウェアに侵されたとき、当然のことながらハードディスクも無防備になってしまいます。このテーマでは、MBR やシステムファイルなど重要なストレージ領域を保護したり、それら重要なファイルや OS カーネルをを狙うマルウェアを検知する、仮想化技術をベースとした侵入検知システムの開発をおこないます。メモリダンプやデバイスドライバレベルの低いレイヤーの情報を使ってマルウェアを探し当てたり、OS の重要な部分を守ることはそれなりに難しいですが、是非チャレンジしてみてください。このテーマで得られるいろいろな知識と実践経験は、今後技術的なマルウェア対策を趣味にしたい人にとって、よいベースになるでしょう。

また、仮想化技術やストレージ周りに興味がある人はもちろん、OS のメモリダンプやファイルシステムの構造などに興味がある人にもオススメのテーマです。

テーマ 06: CTF サーバの自動監視運用システムの開発

CTF（Capture the flag）では、サーバへの侵入を前提とした問題などがよく出されます。運営側にとってはサーバが不安定になったり、異常な過負荷状態になることは避けたいことです。しかし、ユーザーのいたずらや悪意のない純粋な攻撃によって、しばしばそういう状況が発生します。

このテーマでは、そういった侵入を前提とした CTF における、サーバの異常検知と原因特定を自動的に行なうシステムの開発にチャレンジしてもらいます。なので、CTF に積極的に参加していて、かつプログラミングにも興味がある人におすすめです。実習内容として、カーネルモジュールや API フックを使った開発をおこないますので、このクラスの中では比較的高いレイヤーのプログラミングになります。

（5）ゼミ名「ルーター自作ゼミ」（ゼミ長：竹迫良範）

■概要

最近話題になっている DNS サーバーの脆弱性や、偽装通信の手法を理解するためには、
まずインターネットでやりとりされている通信パケットの仕組みについて理解して
おく必要があります。

ルーター自作ゼミでは、ルーターのプログラムを C 言語で自作することによって、
インターネットの通信の仕組みについて理解を深めます。発展的な内容として、
一本足ルーターや自作の暗号化方式を使った VPN 通信プログラムを作成します。

■テーマ

・テーマ#01「Raspberry Pi をベースに自作ルーターを作ってみよう」
・テーマ#02「自作の暗号化方式を使った VPN ルーターを作ってみよう」

■応募条件

1. C 言語プログラムの読み書きができること
2. 無線 LAN やブロードバンドルーターの設定をしたことがあること
3. ping、tracert (traceroute) コマンドを実行したことがあること