H2HC MAGZINE ED.7 / MAIO 2014

H2HC

HACKERS TO HACKERS CONFERENCE

MAGAZINE

EXCLUSIVO

O IMPLANTE DO XEM, O CHIP RFID IMPLANTÁVEL!

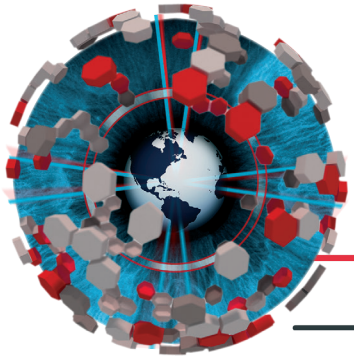
XEM , O CHIP RFID IMPLANTÁVEL!

TUDO SOBRE O CHIP E SUA APLICAÇÃO EM UMA MATÉRIA EXCLUSIVA PARA A H2HC MAGAZINE!

STACK FRAMES

O QUE SÃO, E PARA QUE SERVEM? O ASSUNTO DA NOVA COLUNA POR YGOR PARREIRA E FILIPE BALESTRA

NOVIDADE: H2HC WORLD POR RENEGADOSCAST



H2HC

HACKERS TO HACKERS CONFERENCE

MAGAZINE

H2HC MAGAZINE
EDIÇÃO 7
MAIO DE 2014

Direção Geral
Rodrigo Rubira Branco
Filipe Balestra

Direção de Arte / Criação
Amanda Vieira

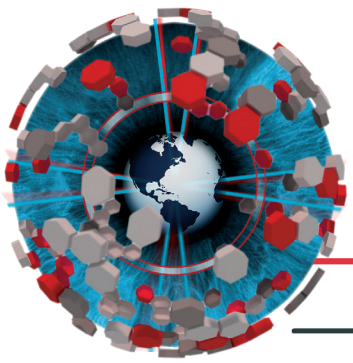
Coordenação Administrativa / Mídias Sociais
Laila Duelle

Redação / Revisão Técnica
Gabriel Negreira Barbosa
Ygor da Rocha Parreira
Jordan Bonagura
Marcelo M. Fleury

Agradecimentos

Equipe do **Renegados Cast**
Kelvin Clark
Pedro Fausto Rodrigues Leite Jr.
Raphael Bastos
Área 31 Hackerspace
Amal Graafstra

Índice



H2HC

HACKERS TO HACKERS CONFERENCE

MAGAZINE

EXCLUSIVO: O implante do XEM, o chip RFID implantável

TEXTO POR LAILA DUELLE
IMAGENS CEDIDAS POR ÁREA 31 HACKERSPACE

A H2HC Magazine acompanhou com exclusividade o implante do biochip XEM!

O profissional de segurança da informação, Raphael Bastos, foi o primeiro brasileiro a implantar o dispositivo. O implante foi feito na cidade de Belo Horizonte, Minas Gerais, no ultimo sábado, dia 03 de maio, na Old Lines Tattoo Shop. O procedimento foi simples e durou cerca de 3 minutos: não é necessário nenhum procedimento cirúrgico para esse fim e ao final só é necessário um pequeno curativo.

Rafael Dias, especialista em body modification, explica que se trata de um procedimento muito simples: com o auxílio de um cateter aplica-se o chip abaixo da pele. O chip tem o tamanho de um grão de arroz, então a perfuração feita para a implantação é mínima, facilitando a cicatrização.

Raphael Bastos disse à nossa redação que no dia seguinte já não tinha nenhum inchaço. Em uma breve comparação, Raphael disse que suas veias da mão eram mais altas que o chip e ao passar a mão o minúsculo dispositivo é quase imperceptível.

Assim que saiu da Old Lines, Raphael foi a seu local de trabalho testar se a praticidade do chip já estava realmente implantada em sua vida. E adivinhem!? Sim, o chip liberou a catraca com facilidade e melhor, o alcance do sinal dele é muito maior do que qualquer cartão magnético de portaria. Já estamos ansiosos aguardando seus próximos testes em seu carro e suas portas.

Mas Raphael não quer parar por aí, este ano ainda pretende implantar o modelo mais novo do chip na outra mão, o XNT, o chip NFC, que tem como função principal liberar o acesso a seu computador. As facilidades que os chips trazem são impressionantes: imagine não ter que carregar mais chaves, cartões e muito menos decorar senhas! É uma facilidade e tanto, mas isso exige muita coragem.

Você pode ver os vídeos exclusivos de como foi todo o processo para a implantação e os testes do chip no link abaixo:

https://www.youtube.com/user/h2hconference

Visando o cenário brasileiro, principalmente o Estado de Minas Gerais que é um Estado muito religioso; o Raphael enfrentou muito preconceito e ataques de religiosos quando divulgou seu projeto ainda antes de implantar o chip.

Nossa equipe contatou Amal Graafstra, autor do livro *"RFID Toys"* e fabricante do chip:

H2HC Magazine: Quando Raphael procurou você para comprar o biochip, o que você pensou em ter o primeiro brasileiro a colocar?

Amal Graafstra: O Brasil é um país muito interessante. O único problema está nas taxas de importação para tecnologias como esta. Muitas pessoas me contataram

com interesse nos chips, mas os custos de transporte e importação são muito altos. O Raphael e o hackerspace Area31 mostraram-se muito interessados nos chips e estamos trabalhando em uma parceria que esperamos que permita mais Brasileiros experimentarem nossos chips num futuro próximo.

H2HC Magazine: O Brasil é um país muito religioso, Raphael e o Area 31 Hackerspace sofreram vários ataques de religiosos . O que você acha disso? É um atraso para a evolução?

Amal Graafstra: Sim, muitas pessoas religiosas disseram que eu era mal, ou que eu estava trabalhando com o demônio. Obviamente a mesma coisa foi dita sobre cartões de crédito e aqui temos um número chamado Número da Segurança Social, o qual diversos grupos religiosos disseram ser do mal na década de 60 quando o mesmo foi criado.

H2HC Magazine: Você tem chips implantados?

Amal Graafstra: Eu implantei dois chips, um em cada mão. Fiz meu primeiro implante em março de 2005 e o segundo em junho do mesmo ano. Minha mão direita teve diversos chips dado que eu removia e substituia diversas vezes. Podem checar no link *http://dangerousthings.com/media-kit* para fotos e informações

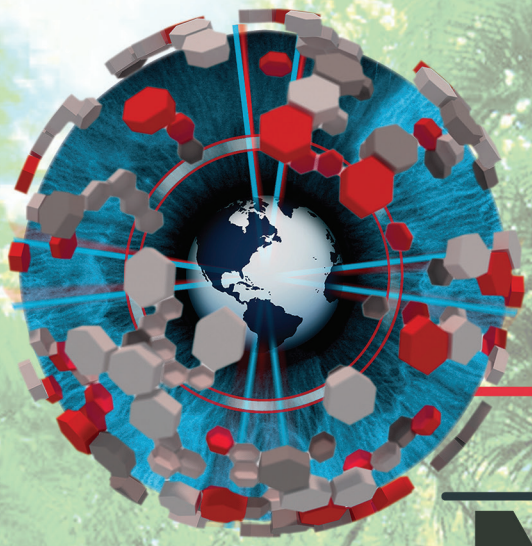
H2HC Magazine: Qual a funcionalidade deles? Para que os usa?

Amal Graafstra: Eu uso meus implantes pra abrir a porta de casa, ligar minha moto, logar em meu computador e abrir um cofre. Também compartilho minha página do facebook com celulares que tenham NFC.

H2HC Magazine: E quando surgiu a ideia de colocar um chip?

Amal Graafstra: Tive essa ideia em 2005 quando tentava atualizar as portas de meu escritório para conseguir entrar sem ter de carregar minhas chaves. Percebi que a tecnologia RFID era barata, fácil de usar e confiável, mas não queria carregar um cartão de acesso comigo. Implantes foram a solução óbvia.

Quer saber mais sobre o assunto? Veja a matéria completa sobre o chip na página 15!



11ª EDIÇÃO 2014

H2HC

HACKERS TO HACKERS CONFERENCE

NA SELVA

18 e 19 de Outubro de 2014
Novotel Morumbi - São Paulo

Você ainda pode garantir seu ingresso com valor promocional!

Dia 14 de maio é o último dia par garantir seu ingresso com valor do 1º lote!

Vá ao nosso site e aproveite as formas de pagamento facilitadas.

Não fique fora dessa! Faça você também parte da história mais antiga da conferência Hacker da América Latina!

Esperamos você em nossa selva! Acesse www.h2hc.com.br e saiba mais.

Engenharia Social
Uma Arma Temida pelas Empresas

POR KELVIN CLARK

Bom, antes de mais nada, vou fazer o que todo mundo faz: *definir o que é engenharia social*. A meu ver, o melhor modo de se definir, seria que a engenharia social é o processo que se emprega para manipular e obter informação de uma pessoa, e pode ser utilizada desde a área militar até a de pentest.

Provavelmente você já ouviu falar de engenharia social. Ela é bem famosa em grande parte por causa do hacker Kevin Mitnick que usou de engenharia social para invadir agências de telecomunicações, de vários filmes como o Prenda-me se for capaz e Jogos de espiões, e de séries como White Collar (Pt-Br Crimes do colarinho branco), sendo esta última uma boa forma de se identificar técnicas discutidas mais a frente.

Assim como um ilusionista faz você prestar a atenção na mão direita dele enquanto a esquerda tira seu relógio, um engenheiro social também se utiliza de técnicas para obter o máximo de informações possíveis e utilizar essas informações que adquiriu para seu próprio bem.

A engenharia social pode ser usada para diversos fins, até mesmo para uso pessoal. Alguns exemplos: persuadir um entrevistador na entrevista de emprego, saber mais sobre um concorrente, descobrir informações e analisar melhor o sexo oposto antes de partir para o flerte.

No caso da área de segurança, a engenharia social também pode ser usada. Alguns exemplos de uso são fazer com que as pessoas forneçam informações sem elas perceberem, manipular as pessoas para lhe dar algum tipo de acesso e colher o máximo de informações possíveis através da observação.

Devido a sua abrangência, a engenharia social acaba então sendo coberta por diversas áreas, como por exemplo: psicológica, militar, judicial e segurança da informação.

Bem, como os engenheiros sociais conseguem essas informações? Quais são os meios que eles usam? Como conseguir tal nível de manipulação?

Simples, os engenheiros sociais usam várias regras implantadas pelo passar dos anos nas cabeças de qualquer pessoa para conseguir o que querem. Verás que, até por ser gentil e educado, um engenheiro social pode se aproveitar de uma pessoa. Para isso eles podem usar: modos de agir, falar e se vestir, beleza, informações (por que não?) para atrair sua confiança, medo, respeito ou até fazer você se sensibilizar por ele.

Existem várias formas de se obter informações de uma pessoa ou empresa usando engenharia social, desde programas como o SET (Social Engineering Tolkit) [1], que vocês podem ler mais a respeito na segunda edição dessa revista, até outros como o Maltego [2]. Redes sociais (como Facebook, LinkedIn, Twitter e outros) também podem ajudar a colher muitas informações, como: trabalho, cargo na empresa, colegas de trabalho, hobbies, lazeres... Isso tudo pode ser correlacionado para melhorar a abordagem a uma pessoa ou empresa.

A seguir vou dar um exemplo de como um engenheiro social poderia entrar dentro de uma empresa que não possua acesso, apenas usando engenharia social.

Por exemplo, responda as seguintes perguntas como se alguém tivesse lhe perguntando em um restaurante.

1-Sabe qual é o prato especial do cardápio de hoje?
2-Costuma almoçar a essa hora? Sabe onde tem um bom restaurante por aqui?
3-Você parece ser legal, seus colegas de trabalho vem almoçar com você?
4-Chefes almoçando junto é sempre ruim, não acha?
5-Será que amanhã vai chover?

A partir disso, vou fazer uma dissecação

dessas perguntas e suas possíveis respostas.

1- A percepção humana está muito mais ativa *no início de uma conversa do que com o decorrer dela. Se uma pessoa na rua pergunta sobre seu trabalho, facilmente irá levantar suspeitas, mas colocando isso em um contexto de uma conversa, irá passar despercebido. Por isso, inicialmente, não se faz as perguntas que realmente interessam.*

2- Perguntar algo mais a fundo que se queira saber e logo depois emendar uma segunda pergunta um pouco mais amigável. Essa técnica é conhecida como Perguntas Mascaradas.

3- Elogios amaciam o ego das pessoas. Logo, com o decorrer da conversa, pode-se fazer elogios para ganhar mais confiança do alvo. Você elogiando faz a pessoa pensar que estão trocando confiança.

4- Pessoas com inimigos em comum tendem a criar certo vínculo, e superiores são inimigos potenciais. Se utilizar disso faz com que se crie um laço em comum com a pessoa, ganhando ainda mais confiança.

5- Não se deve terminar com uma pergunta chave, pois quando essa pessoa se lembrar de você ela vai, provavelmente, se lembrar do último assunto que foi falado. Essa técnica é conhecida como Fechamento.

Com cinco ou mais perguntas um bom engenheiro social consegue informações como horário de almoço, onde costuma almoçar e com quem, se tem colegas de trabalho, quem é o seu superior e se tem uma boa relação com ele.

Pode parecer que são informações irrelevantes, mas para um engenheiro social qualquer informação é importante, e quanto mais informação acumulada melhor.

Bom, sabendo em que horário os funcionários voltam do almoço, será mais fácil eu passar despercebido pela portaria, mas caso eu tivesse que passar pelo porteiro, isso também não seria difícil, como discutido em seguida.

Métodos para se passar pela portaria

1- Se enturmar. Poderia entrar conversando com um "amigo" da empresa que acabei de conhecer no almoço. Desta forma, dificilmente um porteiro pararia para perguntar quem sou, já que estou junto com funcionários que ele já conhece.

2- Tailgating: essa é uma prática de seguir uma pessoa que tem acesso e ir pegando a porta aberta logo depois que o usuário autorizado passar.

3- Educação: muitas vezes as pessoas são educadas, gentis umas com as outras, e podemos nos utilizar dessa gentileza para nosso próprio bem. De que forma? Simples, entre com uma bandeja cheia de copos de café (qualquer empresa precisa de café); por querer ser educado, muitas vezes o segurança ou qualquer funcionário ao ver sua situação, com as mãos cheias, vai querer abrir a porta com todo prazer.

"Pode parecer que são informações irrelevantes, mas para um engenheiro social qualquer informação é importante, e quanto mais informação acumulada melhor"

Pronto, já está dentro, viu que facilidade? Por isso muitas pessoas se utilizam dessa prática para atacar empresas ou corporações, e esse é apenas um dos vários possíveis usos da engenharia social.

Como ser um engenheiro social

Esteja bem arrumado: uma das técnicas de engenharia social é se passar por um superior da pessoa. Pense bem, se entrar em um prédio vestido com um jeans e uma camiseta, dependendo da empresa você muito provavelmente será questionado se é realmente funcionário. Mas se estiver com um bom terno e um sapato italiano, o porteiro provavelmente não irá perguntar, por

ficar com medo de ser repreendido caso você seja um funcionário de alto escalão da empresa.

Conheça a si próprio. Como já dizia Sun Tzu: "Se você conhece o inimigo e conhece a si mesmo, não precisa temer o resultado de cem batalhas" [3]. Essa frase fala o por que deve-se conhecer seus pontos fortes e fracos (todo mundo tem os seus). Basta saber usá-los, pois as vezes ficar focado só no alvo é algo ruim. Tem que se concentrar para não passar nervosismo, não se embolar na fala ou deixar de perceber algo.

Saber analisar as pessoas. Uma pessoa analisa outra sem ao menos perceber que está fazendo isso, está no nosso subconsciente. Tente analisar o máximo possível a pessoa de quem quer se aproximar, e tente se antecipar à análise que essa pessoa também fará de você.

Ter paciência. Por muitas vezes tem que se esperar o momento certo de atacar, estudar muito o inimigo antes de tentar qualquer coisa, até por que muitas vezes o engenheiro social já pode estar dentro da empresa, e qualquer erro pode comprometer todo o processo.

Existem diversas áreas de engenheira social.. Uns gostam de aprender sobre cadeados, trancas, passcards, segurança física em geral; outros vão para o lado de extração de informações ou ainda para a manipulação emocional.

Um dos meios de se evitar ataques de engenharia social é fazer um treinamento com os funcionários. Porém, por muitas vezes, esse treinamento é dispendioso pois demanda horas de trabalho e certo custo financeiro. Adicionalmente, pode-se investir em segurança física para tentar identificar e barrar os engenheiros sociais de adentrar na empresa.

Entretanto, por mais que investimentos sejam aplicados contra engenharia social, sempre se esbarrará em um problema: as pessoas. As pessoas são muito propensas ao erro, e estão a mercê de suas emoções. Elas podem ser manipuladas (até mesmo intencionalmente), e isso faz delas um dos elos mais fracos da segurança da informação, mesmo que treinadas.

Essa dificuldade encontrada em barrar esses tipos de ataque faz com que a engenharia social seja uma arma temidas pelas empresas.

Referências

[1] *SET (Software Engineering Toolkit)*. Acessado em: 17/02/2014)
[2] *Maltego.* (Acessado em: 17/02/2014)
[3] *Trecho retirado do livro A arte da guerra – Sun Tzu* (ISBN 0195014766)

KELVIN CLARK

Tem 20 anos, é estudante de engenharia da computação na faculdade Infnet, estuda e trabalho com linux a 6 anos, é aficionado por tecnologia e segurança da informação, trabalha no Senac-Rio como instrutor do curso técnico de redes, e possui as certificações RHCE, LPIC-3, CompTia Security+ e Network+

Um Fluxo de Design de Componentes Eletrônicos

POR PEDRO FAUSTO RODRIGUES LEITE JR

Resumo: Neste artigo iremos fazer um breve passeio sobre o fluxo de design de componentes eletrônicos (VLSI) e suas diversas etapas. Antes de mais nada, gostaria de salientar que o fluxo que seguiremos aqui está estruturado de um forma mais teórica do que prática. Isto não significa que ele está distante de um design real, de forma alguma. Diversas empresas vão inserindo, modificando ou mesmo removendo algumas destas etapas de acordo com suas necessidades.
Em alguns casos, o fluxo pode ser inteiramente redesenhado e a estrutura se mostrar diferente da aqui apresentada.

Integração em grande escala

Os circuitos integrados, criados através de um processo chamado de VLSI (Very-Large-Scale Integration), são compostos de diversos transistores integrados em um único chip. Para isto, este processo define diversos blocos (metodologia modular) que são dispostos e interconectados de forma otimizada. Com a integração dos circuitos dentro de uma mesma pastilha de silício é possível a redução de área, material, tamanho de trilhas e consequentemente custo. O tempo de transição dos estados elétricos foram reduzidos assim como o tempo de transmissão dos sinais. A complexidade foi escalando, novos problemas aparecendo e hoje existem times inteiros dedicados às pequenas porções do processo [1].

ASIC versus FPGA

Atualmente temos duas principais "tecnologias" (ou escopo) para design de circuitos integrados: o Circuito Integrado de Aplicação Específica (ASIC) e o Arranjo de Portas Programáveis em Campo (FPGA). Conforme referência [3], é de entendimento que o fluxo de design em FPGAs é reduzido (ou um subconjunto em relação ao fluxo de um ASIC). Mas isso não reduz sua complexidade. Para efeito didático, iremos nos ater ao fluxo de design em ASICs.

ASIC: construção e verificação

Um dos pontos mais críticos ao se construir um ASIC é decidir o quanto ele será verificado. Independente do fluxo (e alterações) adotado por uma empresa para um CI, será necessário a verificação funcional, lógica e física do mesmo. Isto irá influenciar na quantidade de erros que o CI poderá ter, bem como seu preço. Por isso neste artigo, ao passarmos pelas etapas de design, iremos passar também pelas etapas de verificação (quando existirem).

O Fluxo

Em uma visão em blocos, o fluxo se apresenta (inicialmente) representado pela figura 1.

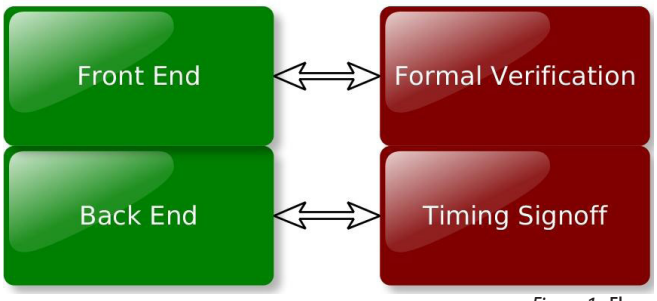


Figura 1: Fluxo

- Front End e Formal Verification: constituem fases cujo objetivo é implementar mediante especificações, modelagem e codificação o componente em questão. Durante algumas destas etapas será realizada a verificação formal do que está sendo implementado. Grosseiramente falando, nesta etapa é realizada a implementação "lógica".

- Back End e Timing Signoff: Nestas etapas o projeto entra em sua implementação "física", sendo os transistores e trilhas dispostos dentro do espaço determinado para o CI. Há determinação de vários outros parâmetros físicos. Concomitantemente, são realizadas simulações nos sinais, tempo de viagem entre blocos, alimentação,

efeitos parasitas, etc.

Vale lembrar mais uma vez que a breve descrição acima pode variar entre fabricantes mas se assemelha em diversos pontos. E para facilitar futuras pesquisas, os termos serão mantidos em inglês. Para diferentes exemplos e termos, favor consultar as referências [2] e [3].
Assim explanado, a figura 2 possui uma imagem dos blocos discutidos, porém expandidos e contemplando "todas" as etapas.

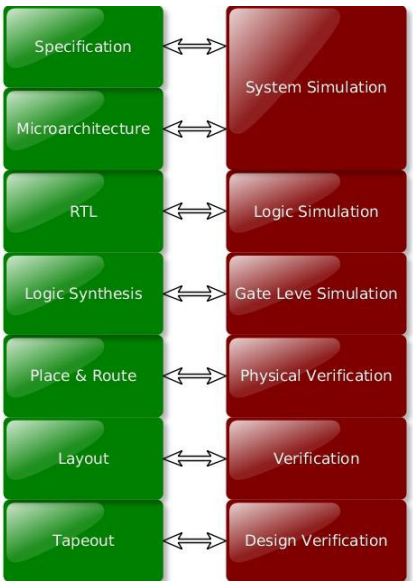


Figura 2: Fluxo Expandido

1. Specification & Microarchitecture

Apesar de serem duas fases distintas, estão intimamente ligadas. De posse da especificação do CI é decidida qual arquitetura será utilizada para atender esta demanda. Esta é uma das partes mais cruciais do projeto. Um erro na escolha da arquitetura

"Um erro na escolha da arquitetura pode ocasionar a falha do projeto inteiro."

pode ocasionar a falha do projeto inteiro.

Neste ponto é possível que uma arquitetura seja reutilizada de outros projetos ou simplesmente seja criada inteiramente do zero. Podem ser realizadas simulações de sistema para estudar o comportamento da arquitetura que será escolhida diante das especificações disponíveis.

2. Register Transfer Level (RTL)

Antes de mais nada, esta etapa pode ser sucedida por (ou suceder-se, em alguns casos) outra etapa conhecida como "Análise Comportamental" (Behavioral Analysis) que, como o próprio nome indica, analisa o comportamento do design. Já o RTL em si contém uma descrição do circuito em termos do fluxo dos sinais entre registradores (registers), fios e operações realizadas através de portas lógicas[4]. Ambas as etapas podem ser descritas através de "linguagens de descrição de hardware" (HDL).

Ao contrário das linguagens de programação, que precisam de inúmeras extensões e adaptações para conseguir descrever um hardware, as HDLs já foram criadas para tal propósito.

Diversas construções algorítmicas que fazem todo sentido em linguagens de software simplesmente são impraticáveis em hardware[5]. As duas HDLs mais conhecidas são: Verilog e VHDL. Para a verificação do que foi codificado com os HDLs é necessário a atuação de uma equipe de verificação lógica que, através de software e metodologias de testabilidade, irão exaustivamente procurar por bugs na implementação.

Quanto maior a cobertura obtida, melhor. É muito importante salientar que esta é outra etapa essencial no projeto.

Ela (em conjunto com as outras etapas de teste) irá influenciar no tempo do projeto, no esforço aplicado, na confiabilidade do hardware e consequentemente no seu custo.

Uma equipe de verificação deve ser, em tese, duas vezes do tamanho da equipe de design. Os esforços necessários pela equipe de verificação são considerados, pelo mercado de fabricantes de ASIC, aproximadamente duas vezes maior do que o necessário pela equipe

de designs[13]. Além disso, o autor sugere que, idealmente, a equipe de verificação não tenha acesso ao HDL. Isto evitaria vícios que prejudicariam a verificação [14] [15].

3. Logic Synthesis

Através das ferramentas de CAD (Computer Aided Design) todo o RTL é traduzido, através da síntese lógica, para um mapeamento que possui a interconectividade entre portas lógicas, fios, entradas, saídas etc, chamado de netlist. Os elementos ali associados estão descritos de forma discreta, sem parâmetros físicos e elétricos. Mais verificações lógicas são realizadas então.

Nesta etapa, a tecnologia que será utilizada é inserida no processo de geração de uma nova netlist. Por tecnologia entende-se o processo de fabricação das pastilhas de silício e dos transistores, que varia com as fábricas (foundries) e fabricantes.

Através da "inserção" dos parâmetros da tecnologia, as simulações lógicas ficarão mais precisas. Os projetistas são capazes de medir os atrasos entre registradores e portas lógicas. Neste ponto, a equipe começa a vislumbrar o verdadeiro comportamento do circuito, quais blocos são maiores, quais são os mais lentos, onde é necessário otimizar e por aí em diante através de uma "análise estática temporal" (Static Time Analysis).

Neste ponto é importante salientar que, a cada etapa do fluxo, teremos uma abstração do design como entrada desta etapa e uma outra abstração como sua respectiva saída. Estas abstrações (ou modelos) podem ser verificadas por uma sub etapa conhecida como "equivalência funcional", que permeia o fluxo. O intuito é garantir de que não houve alteração ou perda das funções implementadas no design [10] [11] [12]. Um exemplo disto seria a comparação entre a netlist e o RTL.

Outro exemplo seria a comparação entre a netlist gerada na síntese lógica e a netlist gerada após a inserção de parâmetros físicos e elétricos. Ainda nesta etapa do fluxo é possível inserir também uma lógica extra que adicionará uma testabilidade maior da lógica do circuito, chamada de Design for Testability (DFT). O DFT é bem diferente do teste funcional, pois testa a observabilidade e controlabilidade de um circuito[6].

4. Place & Route

Nesta fase, o design encontra-se no bloco "Back end" mostrado anteriormente. Agora os componentes são trabalhados efetivamente a nível físico. Um projetista de backend começa a realizar o Floorplan (o equivalente a uma planta baixa) do circuito alocando os blocos, pinos, fontes de alimentação, etc[7]. Os pontos de contato onde serão dispostos os sinais de alimentação são definidos, conforme mostra a figura 3.

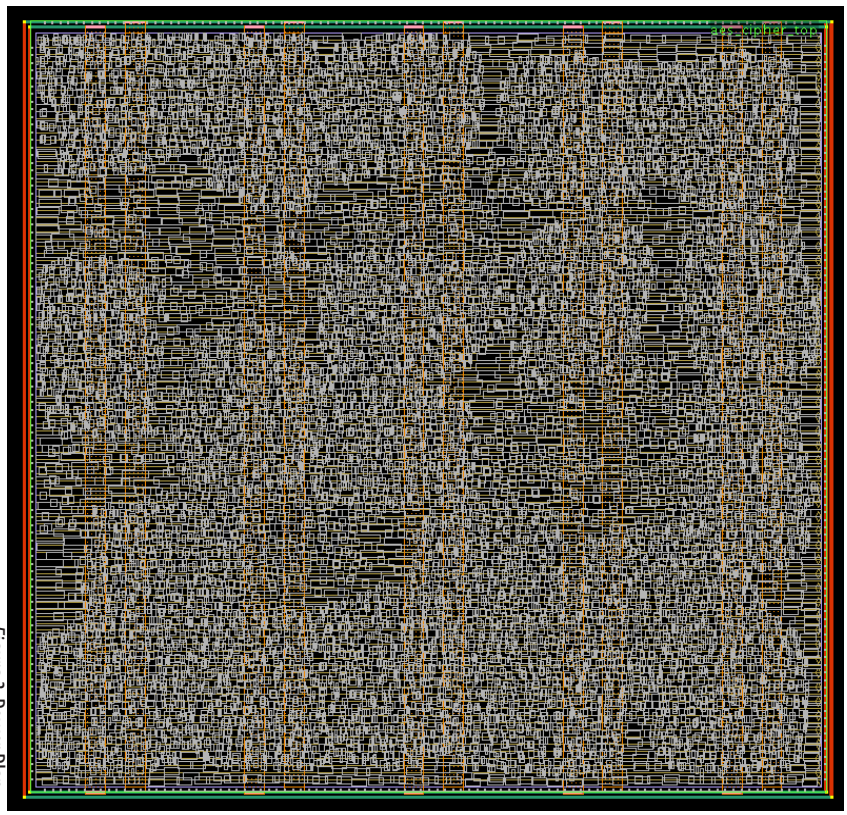


Figura 3: Power Plan

Em seguida as camadas semicondutoras onde os transistores ficarão dispostos (rings, stripes, etc) são inseridas, como exemplifica a figura 4.

Os blocos são distribuídos (ou simplesmente espalhados na área disponível), checados e roteados. A figura 5 é um exemplo.

Há também a inserção do sinal de clock e a otimização do seu caminho para obter os melhores clock path e data paths (exemplificado na figura 6 e visível através das linhas verdes entre transistores), obedecendo as restrições de tempo especificadas.

Além destas fases, há a constante verificação dos parâmetros físicos e das características elétricas do design. Há também a extração de parasitas, decorrentes de efeitos elétricos que começam a surgir em circuitos de tão pequena escala (cross talking, entre outros).

5. Layout

Aqui a equipe é responsável por fazer o layout final que será enviado para as fábricas (foundries) para ser finalmente processado. Ainda há a possibilidade de algumas checagens (DRC). A partir deste ponto, praticamente não há volta. Se existirem erros de projeto, o design será construído errado.

6. Tapeout

Finalmente o design irá para fabricação. Ele é extraído e entregue seguindo um padrão bem estabelecido para representação do mesmo: GDSII[8].

O desenvolvimento de componentes analógicos segue normalmente o mesmo princípio. Em ambos os casos, as equipes realizam suas tarefas interagindo ao máximo para entregar um novo chip para o mercado em tempo hábil. Erros e atrasos na entrega do mesmo podem impor a um projeto, mesmo que simples, uma chance baixíssima de sobreviver no mercado.

Espero que este artigo tenha ajudado a esclarecer, introduzir e quem sabe atrair mais profissionais para este ramo tão fascinante. Na referência [9] há um link para uma apresentação realizada

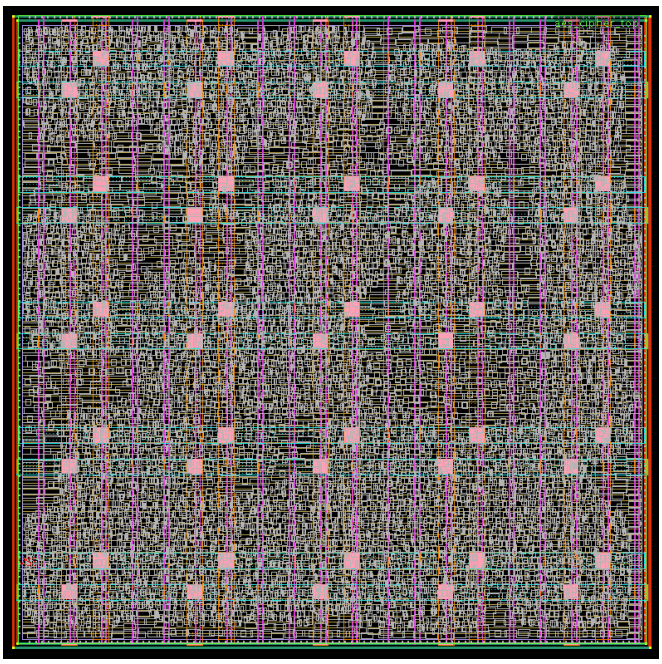


Figura 4: Stripes e rings

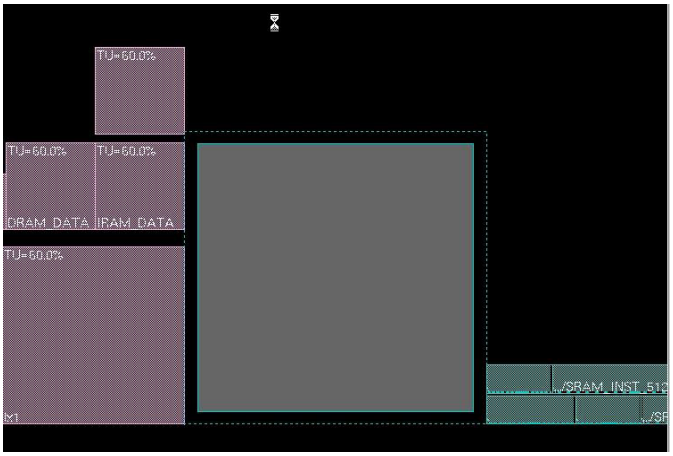


Figura 5: Distribuição dos blocos

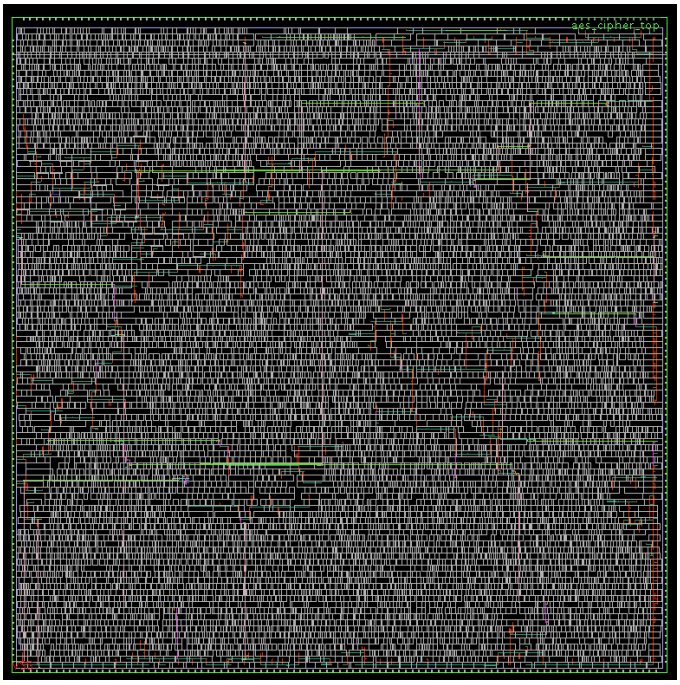


Figura 6: Sinal de clock

pelo Prof. Dr. Jacobus W. Swart, em 2011, sobre alguns projetos, Design Houses e programas de incentivo no Brasil (incluindo o projeto EMC08 no qual o autor teve prazer de participar, referenciado na figura 7).

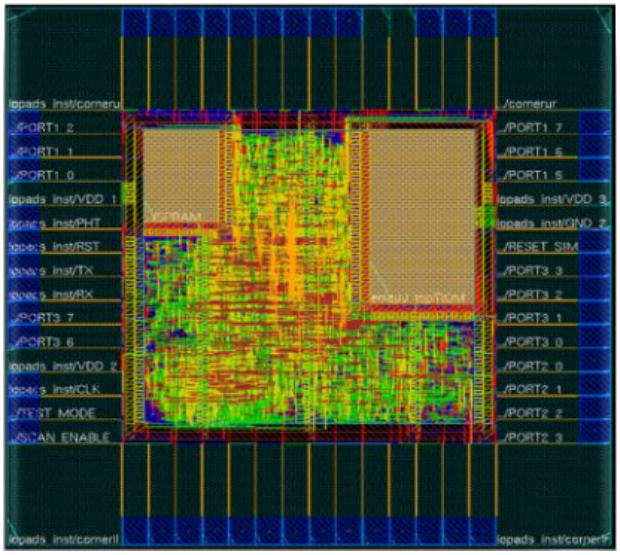


Figura 7: EMC08. Microntrolador de 8 bits

Referências

[1] http://en.wikipedia.org/wiki/VLSI (Acessado em: 10/02/2014)

[2] http://nptel.ac.in/courses/IIT-MADRAS/CAD_for_VLSI_Design_II/magma_tutorial/magma_tutorial.html (Acessado em: 10/02/2014)

[3] http://www.xilinx.com/fpga/asic.htm (Acessado em: 10/02/2014)

[4] http://en.wikipedia.org/wiki/Register-transfer_level (Acessado em: 10/02/2014)

[5] http://en.wikipedia.org/wiki/Hardware_description_language (Acessado em: 10/02/2014)

[6] http://en.wikipedia.org/wiki/Design_for_testing (Acessado em: 10/02/2014)

[7] http://nptel.ac.in/courses/IIT-MADRAS/CAD_for_VLSI_Design_II (Acessado em: 10/02/2014)

[8] http://en.wikipedia.org/wiki/GDSII (Acessado em: 10/02/2014)

[9] http://www.tec.abinee.org.br/2011/arquivos/s202.pdf (Acessado em: 10/02/2014)

[10] http://en.wikipedia.org/wiki/Formal_equivalence_checking (Acessado em: 10/02/2014)

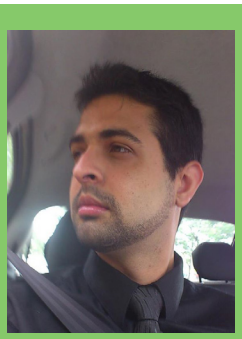
[11] Equivalence Checking of Digital Circuits - Fundamentals, Principles, Methods. Molitor, Paul, Mohnke, Janett. 2004. ISBN 978-1-4020-2603-4

[12] http://asic-soc.blogspot.com.br/2007/10/equivalence-checking-ec.html?m=1 (Acessado em: 10/02/2014)

[13] http://www.synopsys.com/Company/Publications/SynopsysInsight/Pages/Art6-socverif3rdrev-IssQ4-11.aspx?cmp=Insight-I4-2011-Art6 (Acessado em: 10/02/2014)

[14] Comprehensive Functional Verification: The Complete Industry Cycle. Wile, Bruce; Goss, John; Roesner, Wolfgang. 2005. ISBN 978-0127518039

[15] System-on-Chip for Real-Time Applications. Badawy, Wael; A. Julien, Graham. 2002. ISBN 978-1402072543



PEDRO FAUSTO R. LEITE JR.

Pedro teve seus primeiros contatos com componentes eletrônicos ainda na Universidade Federal do Ceará, onde publicou trabalhos e artigos sobre modelagem de dispositivos CMOS e memórias Flash. Trabalhou com SysAdmin, focando em hardening de servidores Linux durante o início de sua carreira profissional. Participante do programa CI Brasil para formação de projetistas. Atuou como projetista na Freescale Semicondutor como parte do programa. Atualmente atua como analista de segurança na Ativas Data Center.


11ª EDIÇÃO 2014
H2HC
HACKERS TO HACKERS CONFERENCE
FALTAM 160 DIAS PARA A PRÓXIMA H2HC !
VOCÊ ESTÁ PRONTO PARA A SUA MAIOR AVENTURA TECNOLÓGICA NA SELVA?
GARANTA JÁ SEU INGRESSO EM NOSSO SITE
www.h2hc.com.br

XEM - O chip RFID implantável

POR RAPHAEL BASTOS

Fonte: www.area31.net.br/wiki/Arquivo:Nfc-xnt04.png

O xEM é uma tag RFID que pode ser implantada no corpo humano. Ela permite que apenas aproximando a mão do leitor, os "cyborgs" destranquem portas, telefones, façam login em computadores, liguem veículos e etc. Também permite compartilhar contatos, vídeos do YouTube, páginas do Facebook com seus amigos apenas lendo o implante, usando um computador, smartphone ou tablet com suporte a RFID.

Nós do Área31 Hackerspace, um hackerspace de Belo Horizonte, inicialmente fizemos testes com os chips usando o hardware opensource Raspberry Pi (ARMv6) e Cubieboard2 (ARMv7) bem como computadores e servidores (x86 e x86_64),
além de criar novos meios de utilização e aplicações integradas. Um dos projetos do hackerspace é o de abertura de um carro utilizando apenas o biochip.

Também estamos sanando problemas de compatibilidade dos leitores NFC e RFID com sistemas operacionais livres, como Linux, FreeBSD e similares. Desde a série 3.x do Linux vem sendo adicionados códigos à árvore oficial do kernel, que possibilitam suporte nativo e automático aos hardwares necessários para este projeto. Estamos bem felizes com a confiabilidade no que se refere a estabilidade e segurança do suporte a NFC e RFID.

Área31 HACKERSPACE

www.area31.net.br

Fonte: www.area31.net.br/wiki/Arquivo:Area31_hackspace_transp.png

Especificações do biochip

O xEM Possui apenas 2 milímetros por 1,2 milímetros cilíndrico. É aproximadamente do tamanho de um grão de arroz. O chip é revestido em vidro Schott 8625 biocompatível. É um tipo comum de vidro usado em dispositivos implantáveis.

Trata-se de um dispositivo passivo, não precisa de bateria e é completamente inerte até que se aproxime do leitor RFID.

- Identificador único e imutável

- Compatível com ISO 14443-A wireless standard

- Cilindro de 2 milímetros por 12 milímetros(2x12) de chumbo borosilicato bio-compatível

- Frequência 125kHZ compatível com TAG EM4200

- O valor aproximado de cada biochip é de de U$33,00 dólares e já está disponível para venda pública

Onde e como é instalado

O local ideal para a instalação é entre a membrana do polegar e o dedo indicador. Como o alcance de leitura é curto, requer que seja localizado na mão, de modo que pode ser facilmente levado para perto dos leitores RFID.

O xEM é pequeno o suficiente para ser instalado por um *piercer* profissional.

Fonte: www.area31.net.br/wiki/Arquivo:Nfc-xnt01.png

Riscos físicos, efeitos colaterais e sobre a cicatrização

De acordo com pessoas que colocaram piercings em locais como a língua , nariz, ou cartilagem da orelha, a dor é similar. O processo normalmente leva cerca de 5 a 10 segundos. Uma vez que o processo for concluído, pode haver pouca ou nenhuma dor por alguns dias. A cicatrização normalmente demora de duas a quatro semanas, mas não há qualquer problema em usar a mão levemente durante a primeira semana.

Como acontece com qualquer procedimento que envolve o corpo há risco, porém é o mesmo risco de se colocar um simples piercing, ou seja, se feito corretamente por um profissional em um ambiente de estúdio limpo o risco é muito baixo. A infecção é o risco mais comum , seguido pela rejeição do biochip. Amal Graafstra, idealizador do projeto, tem dois implantes, um em cada mão desde março de 2005, e até o momento não houveram complicações.

Remoção em casos de dano ao biochip

Qualquer médico familiarizado com cirurgia básica pode facilmente remover o biochip com um pequeno corte de bisturi. Nós também oferecemos guias em vídeo gratuitos para qualquer médico buscando o procedimento de remoção.

Até o momento, mais de 1.000 pessoas já implantaram chips semelhantes(xEM e xM1). Amal Graafstra, idealizador do projeto, teve dois implantes semelhantes desde março de 2005. Em todo esse tempo, ninguém que tenha instalado esses dispositivos no local sugerido - a membrana

entre o polegar e o dedo indicador - relatou qualquer quebra. Uma vez colocado, o dispositivo é bastante resistente. No entanto,caso venha a quebrar, inchaço e leve a dor são esperados. Se isto ocorrer, a qualquer médico de clínica geral deve ser suficientemente habilitado para remover o dispositivo com a ajuda de um anestésico local e um bisturi.

Privacidade e riscos de quebra de sigilo

O tamanho do dispositivo é extremamente pequeno , a gama operacional é muito curta. Devido a esta limitação, seria muito difícil para alguém para "rastrear" o biochip de uma pessoa. A construção de um dispositivo leitor maior e mais poderoso que poderia ler o chip por alguns metros pode ser possível, no entanto, a real possibilidade de construção deste tipo de aparelho é bem remota.

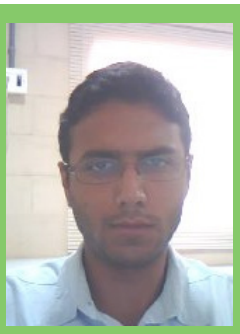
Quem é *Amal Graafstra*, o criador do biochip

Amal tem implantes duplos de RFID, autor do livro RFID Toys, e palestrante na TEDx. Seu interesse em RFID e NFC começou em 2005 como uma solução simples para um problema simples. Ele queria um fácil acesso ao seu escritório. Ele explorou opções biométricas é descobriu que elas eram muito caras, e eram vulneráveis ao vandalismo. A tecnologia de cartão de acesso baseado em RFID foi barato, confiável, e tem uma eficiência maior contra os elementos da natureza e vândalos. Ele optou por mesclar o melhor dos dois mundos e implantar uma tag RFID, em vez de levar um cartão de acesso na carteira. Desde então, ele está escreveu RFID Toys e tem falado sobre RFID e biohacking em diversos eventos, incluindo o Simpósio Internacional de Tecnologia e Sociedade e TEDxSFU.

Fonte: www.area31.net.br/wiki/Arquivo:Nfc-xnt02.jpg

Referências

http://a31.com.br/biochip-rfid
http://a31.com.br/raphaelbastos



RAPHAEL BASTOS

Estudou Engenharia Eletrônica e de Telecomunicação, é um dos autores do SlackBook-ptBR, fundou o Grupo de Usuários Slackware de MG (GUS-MG), realizador dos eventos II Oficina Livre, Slackware Show Brasil e BHACK Conference, desenvolvedor Funtoo Linux, fundador do Área 31 Hackerspace, criador e mantenedor de distribuições Linux para arquiteturas ARM e x86, editor e mantenedor da SlackZine, trabalha atualmente com consultoria de soluções UNIX, infra estrutura de altíssima disponibilidade, capacity planning, replicação, integração de ambientes e/ou sistemas operacionais, segurança da informação, MTA, redes, virtualização, cloud computing e clusters.

FOREWORDS – FUNDAMENTOS PARA COMPUTAÇÃO OFENSIVA

POR YGOR DA ROCHA PARREIRA

E pelo terceiro ano consecutivo, meu treinamento de fundamentos para hacking não aconteceu na H2HC por falta de quórum. Como sempre acontecia nos outros anos, me questionei novamente se o mesmo era importante para alguém que quer aprender como quebrar códigos ou é apenas *bullshit* de alguém que está ficando velho (eu).

Desde o início dos meus estudos no hacking, na maioria dos documentos que lia sempre sentia falta de algo mais, pois me parecia que os documentos estavam incompletos. Até que descobri o que achei ser a chave para a Matrix, que é o conhecimento dos fundamentos da computação. Sim, conhecimento de arquitetura, sistemas operacionais, compiladores, redes, etc. Alguns deles eu já possuía, mas em outros eu era extremamente carente. E foi aí que decidi parar de pesquisar hacking, e focar somente em fundamentos.

Por imaginar que quem pesquisa hacking passa pelos mesmos questionamentos que passei, e não acha facilmente as respostas para estes questionamentos, pensei em compilar tudo isso em um treinamento. Tal treinamento acredito ser algo novo, algo que aproxima mais o mundo acadêmico do mundo hacking. A minha experiência me mostra que o meio acadêmico peca por teorizar demais, e o meio hacker peca por falta da teoria. Na minha visão, o ideal é o caminho do meio que une os dois mundos.
Como os treinamentos não deram certo, a equipe da H2HC Magazine me fez um desafio: Criar e manter uma sessão focada em fundamentos para computação ofensiva, que é o que venho pesquisando nos últimos cinco anos. Acredito tanto que este conteúdo é importante, que ouso fazer um comparativo com uma passagem do filme Matrix onde Morpheus conversa com Neo:

- **Neo**: O que está tentando me dizer? Que posso desviar de balas?
- **Morpheus:** Não, Neo. Estou dizendo que, quando estiver pronto, isso não será necessário.[1]

Mas o que quero dizer com isso? Quero dizer que, quando você compreender bem os fundamentos da computação, o que precisa ser feito para subverter um sistema se torna algo óbvio. Mas então os documentos que explicam as técnicas perdem o sentido? Não. Eles continuam tendo seu valor, pois trazem os macetes para a exploração, mas nada que você não consiga achar com um pouco mais de pesquisa (assumindo que já se possui conhecimento dos fundamentos). Fora que facilita seu entendimento de qualquer técnica que venha a estudar.

Portanto, senhores, os convido e desafio a entrar neste mundo fascinante que é os fundamentos de computação com foco em quebra de sistemas. Cada artigo terá foco em apenas um tópico, onde o(s) autor(es) intenciona(m) esgotar tudo sobre o mesmo. Tópicos correlacionados, quando necessários, serão citados assumindo prévio conhecimento do leitor. O objetivo aqui não é dificultar a leitura do artigo, mas sim dar completude ao tópico central. Com o tempo, escreverei(emos) sobre tais tópicos correlacionados utilizando esta mesma abordagem.

O objetivo maior é criar diversos artigos detalhados, que se interconectam e formam uma boa base de computação para a parte ofensiva da coisa. Portanto, fiquem tranquilos se encontrarem um termo ou citação de uma técnica que ainda não conhecem, pois posteriormente escreverei(emos) sobre ela(s). No início vai ser trabalhoso, mas com o tempo o quebra-cabeça vai se completando.

Os artigos sempre terão uma abordagem científica prática, e quando possível virão recheados de dicas e truques. Os mesmos serão escritos primariamente por mim, Ygor (dmr), mas sempre que possível também terá outro autor. O objetivo aqui é diminuir os erros, e dar riqueza de detalhes e truques de acordo com o assunto abordado. Mesmo o artigo sendo escrito por um revisor (eu), ele passa pelo processo normal de revisão e aprovação de todo o time de revisão.

Um ponto importante a enfatizar é que, o(s) autor(es) e a revista tem compromisso com a corretude das informações apresentadas, e almejam que a didática seja boa o suficiente para que os leitores aprendam completamente o tema. Portanto, pedimos que sempre que possível nos encaminhem dúvidas, erros e dicas sobre a didática empregada.

No mais, Wake up Neo, e sejam bem vindos. Aproveitem esta viagem!

[1] - *Tradução livre*

Stack Frames - O Que São e Para Que Servem?

POR YGOR DA ROCHA PARREIRA E FILIPE BALESTRA

Este artigo está dividido em duas partes, onde a primeira fala sobre o funcionamento da *stack*, e a segunda fala sobre os *stack frames*. As duas partes abusam de exercícios práticos como forma de comprovar os conceitos apresentados.

0x0 - Introdução

O processo de resolução de problemas usando computadores passa pela definição de um algoritmo, que posteriormente é concretizado em um programa. Estes problemas frequentemente são decompostos em problemas menores, de forma a facilitar sua resolução. Usualmente nas linguagens de programação estes problemas menores são compartimentados em procedimentos e/ou funções, onde o que difere estes é o fato de que os procedimentos não retornam valor, mas as funções sim.

A maioria das linguagens de programação suporta o conceito de procedimentos e funções que possuem variáveis locais. Estas variáveis não podem ser armazenadas em endereços fixos da memória, pois em caso de chamadas recursivas ao código elas seriam sobrescritas perdendo assim o valor anterior. A solução adotada pela maioria das arquiteturas é o uso de uma estrutura de dados conhecida como pilha *(stack)* para armazenar tais variáveis.

Stack Frames são estruturas criadas no segmento de *stack* contido no *address space* do processo, para armazenar os argumentos a serem passados para outra função/procedimento[1] que por ventura venha a ser chamada, variáveis locais da função/procedimento corrente e informações que garantam que o programa consiga retomar o processamento de onde parou no instante em que o procedimento corrente foi chamado.

Este artigo mostra como os sistemas computacionais utilizam a estrutura de *stack* provida pelo *hardware*, para controlar o contexto de funções e procedimentos nos processos. Para isto adotou-se para os exemplos a arquitetura *IA-32 (Intel Architecture 32 bits)*, o sistema operacional *GNU/Linux Debian 7.3* com seu conjunto de ferramentas, sendo elas: *gcc v4.7.2-5, binutils v2.22-8, kernel v3.2.0-4-686-pae, strace v4.5.20-2.3, gdb v7.4.1+dfsg-0.1, ltrace v0.5.3-2.1, file v5.11-2, debianutils v4.3.2, coreutils v8.13-3.5 e bc v1.06.95-2.* Para facilitar o entendimento, as configurações de randomização de endereços[2] foram desabilitadas durante a execução dos testes.

0x1 – Funcionamento da *Stack*

Stack é uma estrutura de dados na qual as operações de inserção e remoção de itens são executados por uma única extremidade denominada topo. Ela implementa uma política onde o ultimo a entrar é o primeiro a sair (*LIFO - last-in, first-out*). A operação de inserção é executada pela instrução ***push*** e a de remoção é executada pela instrução ***pop***[3]. Em um Linux sem randomização de endereços, este segmento fica alocado a partir do endereço 0xBFFFFFFF, e na IA-32 ele cresce para os endereços baixos, Figura 1.

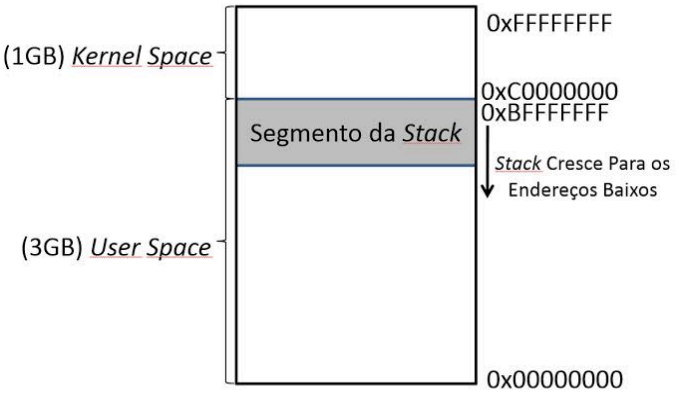


Figura 1 – Localização do segmento de stack dentro do address space de um processo no Linux

[1] - *Isso pode variar de acordo com Calling Convention, ABI e Arquiteturas.*
[2] - *Ver sobre ASLR (Address Space Layout Randomization - http://en.wikipedia.org/wiki/Address_space_layout_randomization).*
[3] - *Existem outras formas de se colocar e remover dados na stack. A Tabela 1 mostra instruções equivalentes que fazem a mesma coisa que as instruções citadas aqui.*

A Listagem 1 mostra onde o segmento de *stack* fica alocado no *address space* do processo durante a execução do programa **/bin/ls**.

```
root@research:~# echo 0 > /proc/sys/kernel/randomize_va_space // Desabilita a randomização de endereços.
root@research:~# which ls // Localiza onde está o binário do programa ls.
/bin/ls
root@research:~# file /bin/ls // Confirma se realmente é um binário ELF executável.
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/
Linux 2.6.26, BuildID[sha1]=0xd3280633faaabf56a14a26693d2f810a32222e51, stripped
root@research:~# gdb -q /bin/ls // Inicia o gdb (GNU Debugger) carregando o binário /bin/ls.
Reading symbols from /bin/ls...(no debugging symbols found)...done.
(gdb) break main // Tentativa frustrada de se criar um break-point no início do código do programa.
Function "main" not defined.
Make breakpoint pending on future shared library load? (y or [n]) n
(gdb) shell ltrace ls 2>&1 | head -n 1 // Localiza o endereço de início do código do programa.
__libc_start_main(0x8049f30, 1, 0xbffffd24, 0x805b760, 0x805b750 <unfinished ...>
(gdb) break *0x8049f30 // Cria um break-point no início do código do programa.
Breakpoint 1 at 0x8049f30
(gdb) r // Executa o programa.
Starting program: /bin/ls
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/i686/cmov/libthread_db.so.1".
Breakpoint 1, 0x08049f30 in ?? ()
(gdb) info inferiors // Verifica qual o PID (Process ID) do processo em execução.
  Num  Description       Executable
* 1    process 2508      /bin/ls
(gdb) shell grep stack /proc/2508/maps // Verifica o mapeamento da stack dentro do address space do processo.
bffdf000-c0000000 rw-p 00000000 00:00 0          [stack]
(gdb) kill // Encerra a execução do programa.
Kill the program being debugged? (y or n) y
(gdb) quit
root@research:~#
```

Listagem 1 – Verificação do mapeamento do segmento de stack durante a execução do programa /bin/ls

Após desabilitar a randomização de endereços, busca-se onde está localizado o programa **ls**. Em seguida é verificado se o arquivo é um binário ELF executável. Aqui cabe notar que se trata de um binário ELF executável onde foi removido todas as informações que poderiam auxiliar no processo de *debug* (*stripped*), como os símbolos que informam o nome das funções e informações de *debug*.

O gdb carrega o binário e disponibiliza um *prompt* de comando. Em uma tentativa inútil, tenta-se criar um *break point* no início do que seria a execução do código do programa (*break main*), onde o gdb prontamente informa que este símbolo não foi definido neste programa (*Function "main" not defined.*). Este comportamento se deve ao binário estar *stripped*.

Como alternativa, sabe-se que o ponto de entrada (*entry-point*) da execução de código na maioria dos programas ELF no Linux não aponta para o início do código do programa (*função main()*), mas para um ponto em seu segmento *.text* que chamará a rotina de inicialização da *LibC* responsável por inicializar o ambiente de execução para o processo, e chamar o código do program[4]. Quando a rotina de inicialização da LibC é chamada, o primeiro parâmetro dela é o endereço do início do código do programa propriamente dito.

[4] - *Ver a primeira questão da sessão 0x2 – Perguntas Mais Frequentes.*

Desta forma foi possível encontrar o endereço da função *main()* e criar um *break-point* nele, para que o gdb interrompesse a execução logo no início do código do programa.

Após o gdb começar a executar o programa *ls*, ele logo para sua execução no *break-point* criado. Neste ponto obtêm-se o PID (*Process ID*) do processo criado de forma a recuperar a faixa de endereços onde o segmento de *stack* foi mapeado, a partir do */proc/<pid>/maps*. A *stack* foi mapeada na faixa de endereço de 0xbffdf000 até 0xc0000000 (não inclusivo)[5]. Como na arquitetura *Intel* a *stack* cresce para os endereços baixos, é correto dizer que ela foi mapeada a partir do endereço 0xc0000000 (não inclusivo) até o endereço 0xbffdf000.

Fazendo um rápido cálculo nota-se que este intervalo equivale a 132KB, o que faz sentido, pois normalmente o Linux executando na arquitetura IA-32 utiliza páginas de memória em tamanhos de 4KB[6], que é o mais comum nesta arquitetura. Tal cálculo é mostrado na Listagem 2. O fato de se ter um segmento de 132KB mapeado, não significa que todo ele esteja em uso, mas sim que este espaço está disponível para ser utilizado pelo processo. Exemplo: Você pode ter um segmento de 132KB mapeado, mas estar usando apenas 512 bytes, Figura 2.

```
root@research:~# bc -q
ibase=16 // Entrada em base 16.
C0000000-BFFDF000 // Cálculo.
135168 // Resultado em bytes – Saída em decimal.
ibase=A // Entrada em base 10.
135168/1024 // Converte de bytes para kilobytes.
132 // Resultado em kilobytes.
quit
root@research:~#
```

Listagem 2 – Calculando o tamanho do segmento de stack durante a execução do programa /bin/ls

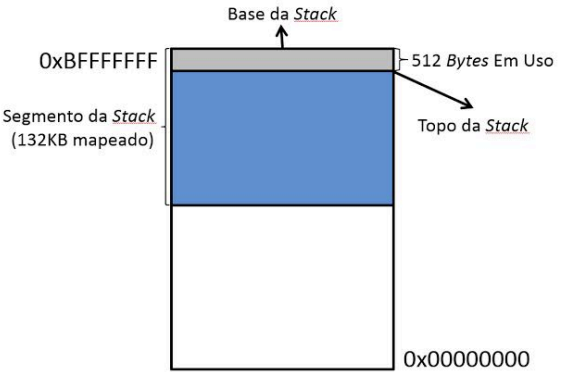


Figura 2 – Uso parcial da stack mapeada dentro do address space de um processo no Linux

Na IA-32, o registrador ESP *(Extended Stack Pointer)* atua como um ponteiro que sempre aponta para o topo da *stack*. As instruções **push** e **pop** trabalham implicitamente com este ponteiro. O código da Listagem 3 é analisado passo a passo na Listagem 4 como forma de demonstrar o funcionamento da *stack*.

```
root@research:~/H2HC_Magazine/Fundamentos/Stack_Frame# cat how_stack_works.s
# To Mount:    as -o how_stack_works.o how_stack_works.s
# To Link:     ld -o how_stack_works how_stack_works.o
.text
.globl _start
_start:
     pushl  $0x4   # Coloca literal no topo da stack.
     pushl  $0x8   # Coloca literal no topo da stack.
     pushl  $0xc   # Coloca literal no topo da stack.

     popl   %eax   # EAX recebe literal do topo da stack.
     popl   %ebx   # EBX recebe literal do topo da stack.
     popl   %ecx   # ECX recebe literal do topo da stack.
```

[5] - *Ver a segunda questão da sessão 0x2 – Perguntas Mais Frequentes.*
[6] - *Existem casos em que é utilizado páginas de 4MB.*

```
    # exit(0);
    xorl   %eax, %eax
    xorl   %ebx, %ebx
    incl   %eax
    int    $0x80
root@research:~/H2HC_Magazine/Fundamentos/Stack_Frame# as -o how_stack_works.o how_stack_works.s
// Monta o fonte assembly.
root@research:~/H2HC_Magazine/Fundamentos/Stack_Frame# ld -o how_stack_works how_stack_works.o //
Gera o binário executável a partir do código objeto criado pelo montador as (GAS) .
root@research:~/H2HC_Magazine/Fundamentos/Stack_Frame# file how_stack_works
how_stack_works: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, not stripped
root@research:~/H2HC_Magazine/Fundamentos/Stack_Frame#
```

Listagem 3 – Programa how_stack_works que demonstra o funcionamento da stack

```
root@research:~/H2HC_Magazine/Fundamentos/Stack_Frame# gdb -q how_stack_works
Reading symbols from /root/H2HC_Magazine/Fundamentos/Stack_Frame/how_stack_works...(no debugging
symbols found)...done.
(gdb) break _start // Cria um break-point no início do código.
Breakpoint 1 at 0x8048054 // Endereço do início do código.
(gdb) r  // Executa o programa.
Starting program: /root/H2HC_Magazine/Fundamentos/Stack_Frame/how_stack_works

Breakpoint 1, 0x08048054 in _start ()
(gdb) x/6i 0x8048054   // Visualiza as primeiras seis instruções do programa.
=> 0x8048054 <_start>:  push   $0x4 // O break-point está aqui, vide indicação da seta.
   0x8048056 <_start+2>:     push   $0x8
   0x8048058 <_start+4>:     push   $0xc
   0x804805a <_start+6>:     pop    %eax
   0x804805b <_start+7>:     pop    %ebx
   0x804805c <_start+8>:     pop    %ecx
(gdb) x/3x $esp // Visualiza os três primeiros valores da stack, a partir do topo.
0xbffffc90:    0x00000001     0xbffffdb1     0x00000000 // ESP = 0xbffffc90.
(gdb) i r $eax $ebx $ecx // Estado inicial dos registradores EAX, EBX e ECX.
eax            0x0     0
ebx            0x0     0
ecx            0x0     0
(gdb) ni // Executa a próxima instrução (next instruction).
0x08048056 in _start ()
(gdb) x/6i 0x8048054
   0x8048054 <_start>:  push   $0x4
=> 0x8048056 <_start+2>:     push   $0x8
   0x8048058 <_start+4>:     push   $0xc
   0x804805a <_start+6>:     pop    %eax
   0x804805b <_start+7>:     pop    %ebx
   0x804805c <_start+8>:     pop    %ecx
(gdb) x/3x $esp
0xbffffc8c:    0x00000004     0x00000001     0xbffffdb1 // Stack Pointer decrementado e valor colocado no topo (Na
IA-32 a stack cresce para os endereços baixos).
(gdb) ni
0x08048058 in _start ()
(gdb) x/6i 0x8048054
   0x8048054 <_start>:  push   $0x4
```

```
  0x8048056 <_start+2>:     push  $0x8
=> 0x8048058 <_start+4>:     push  $0xc
  0x804805a <_start+6>:     pop   %eax
  0x804805b <_start+7>:     pop   %ebx
  0x804805c <_start+8>:     pop   %ecx
(gdb) x/3x $esp
0xbffffc88:   0x00000008    0x00000004    0x00000001 // Stack Pointer decrementado e valor colocado no topo.
(gdb) ni
0x0804805a in _start ()
(gdb) x/6i 0x8048054
  0x8048054 <_start>:  push  $0x4
  0x8048056 <_start+2>:     push  $0x8
  0x8048058 <_start+4>:     push  $0xc
=> 0x804805a <_start+6>:     pop   %eax
  0x804805b <_start+7>:     pop   %ebx
  0x804805c <_start+8>:     pop   %ecx
(gdb) x/3x $esp
0xbffffc84:   0x0000000c    0x00000008    0x00000004 // Stack após os 3 "push".
(gdb) i r $eax $ebx $ecx // Antes de remover os valores, conferimos o conteúdo dos registradores que os receberão.
eax         0x0     0
ebx         0x0     0
ecx         0x0     0
(gdb) ni
0x0804805b in _start ()
(gdb) x/6i 0x8048054
  0x8048054 <_start>:  push  $0x4
  0x8048056 <_start+2>:     push  $0x8
  0x8048058 <_start+4>:     push  $0xc
  0x804805a <_start+6>:     pop   %eax
=> 0x804805b <_start+7>:     pop   %ebx
  0x804805c <_start+8>:     pop   %ecx
(gdb) x/3x $esp
0xbffffc88:   0x00000008    0x00000004    0x00000001 // Valor do topo removido e Stack Pointer incrementado.
(gdb) i r $eax $ebx $ecx
eax         0xc     12 // Valor armazenado no registrador EAX.
ebx         0x0     0
ecx         0x0     0
(gdb) ni
0x0804805c in _start ()
(gdb) x/6i 0x8048054
  0x8048054 <_start>:  push  $0x4
  0x8048056 <_start+2>:     push  $0x8
  0x8048058 <_start+4>:     push  $0xc
  0x804805a <_start+6>:     pop   %eax
  0x804805b <_start+7>:     pop   %ebx
=> 0x804805c <_start+8>:     pop   %ecx
(gdb) x/3x $esp
0xbffffc8c:   0x00000004    0x00000001    0xbffffdb1
(gdb) i r $eax $ebx $ecx
eax         0xc     12
ebx         0x8     8 // Valor armazenado no registrador EBX.
ecx         0x0     0
(gdb) ni
```

```
0x0804805d in _start ()
(gdb) x/7i 0x8048054
  0x8048054 <_start>:  push   $0x4
  0x8048056 <_start+2>:     push   $0x8
  0x8048058 <_start+4>:     push   $0xc
  0x804805a <_start+6>:     pop    %eax
  0x804805b <_start+7>:     pop    %ebx
  0x804805c <_start+8>:     pop    %ecx
=> 0x804805d <_start+9>:     xor    %eax,%eax
(gdb) x/3x $esp
0xbffffc90:    0x00000001     0xbffffdb1     0x00000000 // Todos os valores removidos da stack, voltando ao estado
inicial (antes dos "push" iniciais).
(gdb) i r $eax $ebx $ecx
eax            0xc      12
ebx            0x8      8
ecx            0x4      4 // Valor armazenado no registrador ECX.
(gdb) i r $esp
esp            0xbffffc90      0xbffffc90 // Conteúdo do registrador ESP – Topo da stack.
 (gdb) x/6x $esp-0xc // Visualiza as três words abaixo do topo da stack. O topo (ESP) aponta para o valor 0x00000001.
0xbffffc84:    0x0000000c     0x00000008     0x00000004     0x00000001
0xbffffc94:    0xbffffdb1     0x00000000
(gdb) c // Continua a execução sem mais interrupções.
Continuing.
[Inferior 1 (process 2074) exited normally]
(gdb) q
root@research:~/H2HC_Magazine/Fundamentos/Stack_Frame#
```

Listagem 4 – Execução passo a passo do programa how_stack_works para demonstrar o funcionamento da stack

Uma coisa interessante a se notar é que, mesmo quando dados são removidos do topo da *stack* (apenas o ESP é incrementado), estes valores continuam na memória, ou seja, o processador apenas disponibiliza o espaço antes utilizado para uso posterior. Para facilitar a visualização do que acontece com a stack e os registradores durante a execução do programa *how_stack_works*, a Figura 3 mostra as mudanças após a execução de cada linha.

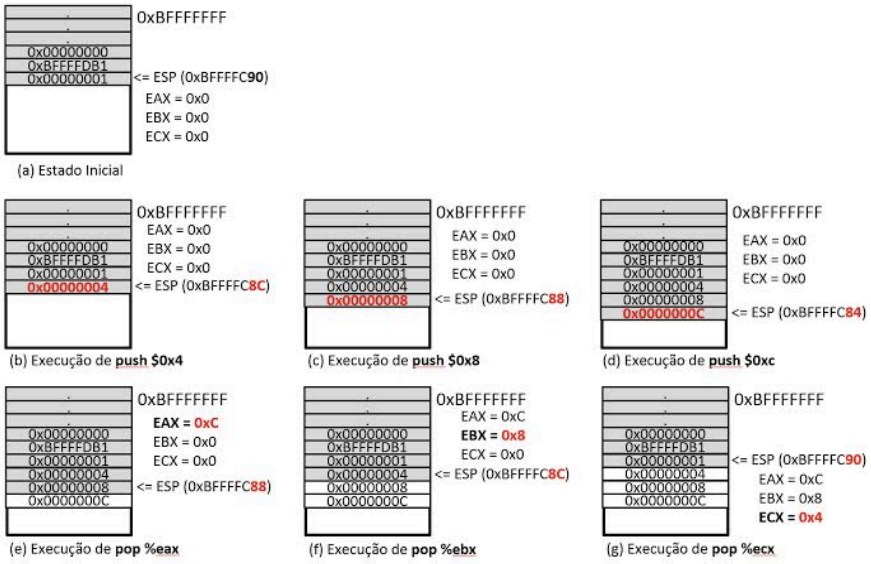


Figura 3 – Estado de registradores, e do segmento de stack durante a execução do programa how_stack_works

Com base no entendimento do funcionamento da *stack*, a Tabela 1 mostra um quadro de equivalência de instruções. As instruções que este artigo trabalha estão na sintaxe AT&T, onde a direção da operação é da esquerda para direita (primeiro a origem e depois o destino), valores precedidos de '$' são interpretados como literais e os registradores são precedidos de '%'. Aqui é possível utilizar sufixos em algumas instruções para indicar qual a quantidade de dados que a operação trabalha, como 'l' para 4 *bytes (long),* 'w' para 2 *bytes (word)* e 'b' para 1 *byte (byte)*.

Instrução	Equivale a:
push $0xc	sub $0x4, %esp mov $0xc, (%esp)
pop %eax	mov (%esp), %eax add $0x4, %esp

Tabela 1 – Equivalência de Instruções

0x2 – Perguntas Mais Frequentes

Questão 1: Vocês disseram que o *entry-point* da maioria dos programas do *Linux* não aponta para o inicio do código do programa propriamente dito (função *main()*), mas sim para um ponto no segmento *.text* do programa, que chama a rotina de inicialização da *LibC*. Porque isso não ocorre com todos os programas?

Resposta 1: Este uso das rotinas de inicialização da *LibC* é comumente acrescentado ao binário executável ELF pelo GCC, mas se você criar um programa em outra linguagem provavelmente o binário será criado de forma diferente.

Um exemplo que mostra isso é o programa em *assembly* criado para demonstrar o uso da *stack* (*how_stack_works*.s), onde nem o montador (as) e nem o *linker* (ld) acrescentou ao binário a chamada ao código desta rotina.

Q 2: O exemplo que mostra o mapeamento da *stack* do processo criado pelo binário ***/bin/ls*** diz que a *stack* começa no endereço 0xc0000000, mas as imagens dos outros exemplos informam que ela começa no endereço 0xbfffffff. Afinal, qual é o correto?

R 2: O endereço 0xc0000000 pertence ao espaço de endereçamento do *kernel*, e não pode ser acessado diretamente por um programa de usuário. Este endereço aparece no */proc/<pid>/maps* porque é não inclusivo.

Q 3: No meu ambiente o */proc/<pid>/maps* não mostra a *stack* começando no endereço 0xC0000000 (não inclusivo). O que estou fazendo de errado?

R 3: Provavelmente você esqueceu de desabilitar o recurso de randomização de endereços no *kernel* do *Linux*, desabilite com o comando *echo 0 > /proc/sys/kernel/randomize_va_space* e seja feliz :)

Este esquema de randomização foi criado para dificultar explorações de corrupção de memória. A idéia original surgiu inicialmente no projeto *PaX*[7] com o nome de *Address Space Layout Randomization* (ASLR), e posteriormente criou-se outra implementação para a *main line* do *kernel* do *Linux*, mas isto é assunto para um outro artigo.

Q 4: Ok, configurei o *kernel* para não randomizar e a *stack* está começando no 0xC0000000 (não inclusivo), mas os endereços não estão batendo com os apresentados no artigo. O que pode ser?

[7] - *https://pax.grsecurity.net/*

R 4: Diversas são as variáveis que influenciam nos endereços empregados por um determinado programa. Assumindo que o código fonte seja o mesmo e a randomização esteja desativada, ainda podem influenciar a versão do compilador, opções de compilação, variáveis de ambiente definidas e outros.

O importante aqui é entender o conceito demonstrado, mas se você utilizar a mesma versão da distribuição *Linux*, compilador, pacote *binutils*, demais pacotes indicados no início, opções de compilação utilizadas neste artigo e acessar via cliente *putty* de *SSH*[8], você deve obter os mesmos endereços.

Q 5: Se todos os programas têm a *stack* mapeada no mesmo endereço, e o *Linux* é um sistema multi-tarefa, não pode ocorrer de um processo modificar a *stack* de outro processo, dado que estão todas na mesma faixa de endereços?

R 5: Assim como qualquer sistema operacional moderno, o *Linux* virtualiza os recursos de memória. Os processos não referenciam diretamente endereços da memória física, em vez disso, o *kernel* atribui a cada processo um *address space* virtual diferente. Ou seja, o mesmo endereço de memória (por exemplo, 0xBFFFFFFF) em processos diferentes referencia diferentes posições na memória física.

---

[8] - *As variáveis de ambientes são diferentes quando se acessa localmente (console) das existentes quando se acessa via SSH.*

YGOR DA ROCHA PARREIRA

Ygor da Rocha Parreira faz pesquisa com computação ofensiva, trabalha com testes de intrusão na empresa DFTI, e é um cara que prefere colocar os *bytes* à frente dos títulos.

FILIPE BALESTRA

Filipe Balestra é especialista em segurança da informação, onde está envolvido desde 1997. Executou projetos de segurança em diversas empresas, é organizador do H2HC além de editor desta revista.
Publicou falhas de segurança em importantes *software*, bem como artigos em locais como Hakin9 e Phrack Magazine.

PAPO B1NÁR10

EM BREVE!

O Renegados Cast surgiu em 2012 com a intenção de gravar um simples Podcast. Dois anos depois e mais de 80 programas, unindo determinação e modernidade, apresentamos conteúdo todos os dias e um Podcast inédito toda segunda feira.

Após o sucesso obtido na *Campus Party 2014*, tivemos a honra de fechar a parceria com a ***H2HC*** para apresentar um conteúdo descontraído e informal para seus leitores.

Boa leitura e grite com a gente: **AVANTE RENEGADOS.**

Sessão Renegada

ELA

Fonte: http://renegadoscast.com/wp-content/uploads/2014/03/SR64.jpg

Direção: Spike Jonze
Elenco: Amy Adams, Artt Butler, Bill Hader, Brian Johnson, Chris Pratt, Dane White, David Azar, Dr. Guy Lewis, Evelyn Edwards, Gracie Prewitt, James Ozasky, Joaquin Phoenix, Kristen Wiig, Luka Jones, Matt Letscher, May Lindstrom, Melanie Seacat, Nicole Grother, Olivia Wilde, Pramod Kumar, Rooney Mara, Samantha Sarakanti, Scarlett Johansson, Steve Zissis
Roteiro: Spike Jonze
Produção: Megan Ellison, Spike Jonze, Vincent Landay
Edição: Eric Zumbrunnen, Jeff Buchanan
Trilha Sonora: Owen Pallett
Fotografia: Hoyte Van Hoytema
Gênero: Comédia Dramática
País: EUA
Duração: 126 min.
Ano: 2013
Estúdio: Annapurna Pictures
Classificação: 14 anos

SINOPSE

Theodore Twombly (Joaquin Phoenix) é um homem complexo e emotivo que trabalha escrevendo cartas pessoais e tocantes para outras pessoas. Com o coração partido após o final de um relacionamento, ele começa a ficar intrigado com um novo e avançado sistema operacional que promete ser uma entidade intuitiva e única. Ao iniciá-lo, ele tem o prazer de conhecer "Samantha", uma voz feminina perspicaz, sensível e surpreendentemente engraçada. A medida em que as necessidades dela aumentam junto com as dele, a amizade dos dois se aprofunda em um eventual amor um pelo outro. ELA é uma história de amor original que explora a natureza evolutiva — e os riscos — da intimidade no mundo moderno.

CRÍTICA RENEGADA

Para alguns pode parecer um absurdo um ser humano ter uma relação com um computador/*software* (Ta.. os japoneses já fizeram isso), porém o que é mostrado no filme vai além, o filme é de uma simplicidade e sutileza que acaba cativando e você entende a motivação de Theodore para ter esse tipo de relação com Samantha. Deixa eu tentar ilustrar um pouco: é como se o sistema operacional do seu PC particular começasse a conversar diretamente com você, como se uma pessoa… alias.. não só uma pessoa, uma amiga. Uma amiga (o) que poderia te ajudar de uma forma com todas as suas coisas, que você acabaria tendo um certo carinho afetivo por ela. E esse sistema operacional funcionaria como uma pessoa literalmente a ponto de realmente conversar com você. Conversar e mostrar o ponto de vista dela, sem parecer algo automático ou forçado, mas natural, como uma pessoa interessante que você acaba de conhecer. As sacadas de "Samantha" (Scarlett Johansson, sensacional) é de uma naturalidade que assusta, pelo fato de não termos algo "ainda" desse jeito. Joaquin Phoenix também está muito bem nesse filme, o fato dele interpretar com a "voz" de Samantha, suas reações e frustrações, vão mostrando o quanto solitário é o personagem Theodore. O Visual Retrô/HighTech da sociedade relatada no filme também é interessante. Finalizando, o filme deve já ter saído de cartaz nos principais cinemas, mas se tiver curiosidade, ainda deve ter cópias por ai. Ah, é um filme sobre comportamento humano, talvez você não concorde com o final dele, mas realmente mereceu o Oscar de Melhor Roteiro Original. #AVANTE

Por: Alessandro "Bob"Bernard

Assista o trailer:
https://www.youtube.com/watch?v=gvlj2nGczoI

O LOBO DE WALL STREET

http://renegadoscast.com/wp-content/uploads/2014/01/SR59.jpg

Direção: Martin Scorsese
Elenco: Cristin Milioti, Jake Hoffman, Jean Dujardin, Jonah Hill, Justin Wheelon, Kenneth Choi, Kyle Chandler,Leonardo DiCaprio, Margot Robbie, P.J. Byrne, Rob Reiner
Roteiro: baseado em livro de Jordan Belfort, Terence Winter
Produção: Emma Tillinger Koskoff, Joey McFarland, Leonardo DiCaprio, Martin Scorsese, Riza Aziz
Trilha Sonora: Howard Shore

SINOPSE

Durante seis meses, Jordan Belfort (Leonardo DiCaprio) trabalhou duro em uma corretora de Wall Street, seguindo os ensinamentos de seu mentor Mark Hanna (Matthew McConaughey). Quando

Fotografia: Rodrigo Prieto
Gênero: Drama
País: EUA
Duração: 179 min.
Ano: 2013
Estúdio: Appian Way / EMJAG Productions / Red Granite Pictures / Sikelia Production
Classificação: 18 anos

finalmente consegue ser contratado como corretor da firma, acontece o Black Monday, que faz com que as bolsas de vários países caiam repentinamente. Sem emprego e bastante ambicioso, ele acaba trabalhando para uma empresa de fundo de quintal que lida com papéis de baixo valor, que não estão na bolsa de valores. É lá que Belfort tem a ideia de montar uma empresa focada neste tipo de negócio, cujas vendas são de valores mais baixos mas, em compensação, o retorno para o corretor é bem mais vantajoso. Ao lado de Donnie (Jonah Hill) e outros amigos dos velhos tempos, ele cria a Stratton Oakmont, uma empresa que faz com que todos enriqueçam rapidamente e, também, levem uma vida dedicada ao prazer.

CRÍTICA RENEGADA

E ae pessoas!!! Blz? Postando diretamente da CAMPUS PARTY 7!!! Bem, cara, que filme foda que te segura numa boa as 3 horas que te proporciona. Leonardo DiCaprio ta foda no filme inteiro, e que interpretação. A história também te prende, por mais que você, alguma vez na sua vida, nunca tenha se interessado nos bastidores da bolsa de valores, muito mais a americana. Todos os coadjuvantes também tem seu peso equilibrado e suas particulariedades no qual conseguem muito 171 em cima de muita gente, o que faz a empresa do filme criada pelo personagem de DiCaprio se tornar tão foda, que muitas festas e outras coisas acontecem, até para justificar a grana que ganham e sumir com ela pelo fato de ser um ato ilegal nos States... Não entendeu nada? Assista o filme que vale a pena Master!! A, ia me esquecendo: Leonardo DiCaprio merece o Oscar. #AVANTE

Por: Alessandro "Bob"Bernard

Assista o trailer:
https://www.youtube.com/watch?v=pabEtIERlic

NEED FOR SPEED

http://renegadoscast.com/wp-content/uploads/2014/03/SR67.jpg

Direção: Scott Waugh
Elenco: Aaron Paul, Anthony B. Harris, Antoni Corone, Beth Waugh, Biff O'Hara, Cabran E. Chamberlain, Carmela Zumbado, Chillie Mo, Dakota Johnson, Demetrice Jackson, Diezel Ramos, Dominic Cooper, E. Roger Mitchell, Frank Brennan, Han Soto, Harrison Gilbertson, Imogen Poots, Jaden Alexander, Jeff Trink, Jill

SINOPSE

Tobey Marshall (Aaron Paul) herdou do pai uma oficina mecânica, onde, juntamente

Jane Clements, Josh Turner, Kaily Alissano, Kanin Howell, Libby Blanton, Logan Holladay, Mahal Montoya, Mary Ellen Itson, Michael Keaton, Michael Rose, Nick Chinlund, Rami Malek, Ramón Rodríguez, Rick Mischke, Scott Ledbetter, Scott Mescudi, Sir Maejor, Steven Wiig, Valdez Williams, William Grammer
Roteiro: George Gatins, John Gatins
Produção: John Gatins, Mark Sourian
Edição: Paul Rubell, Scott Waugh
Fotografia: Shane Hurlbut
Trilha Sonora: Nathan Furst
Gênero: Drama
País: Estados Unidos
Duração: 131 min.
Ano: 2014
Estúdio: DreamWorks SKG / Reliance Entertainment
Classificação: 12 anos

com sua equipe, modifica carros para que se tornem o mais rápido possível. Além disto, Tobey é um exímio piloto e volta e meia participa de rachas. Um dia, o ex-piloto da Fórmula Indy Dino Brewster (Dominic West) o procura para que Tobey possa concluir um Mustang desenvolvido por um gênio da mecânica que já faleceu. Apesar das divergências entre eles, Tobey aceita a proposta por precisar muito do pagamento oferecido por Dino. O carro é concluído e posteriormente vendido. Entretanto, a velha rixa entre eles faz com que disputem um último racha, que conta ainda com a participação de Pete (Harrison Gilbertson), grande amigo de Tobey. A corrida termina em tragédia devido ao falecimento de Pete. Considerado culpado pela morte, Tobey passa dois anos na prisão. Quando enfim é solto, ele organiza um plano para que possa participar de uma conhecida corrida do submundo onde Dino também correrá.

CRÍTICA RENEGADA

Fiquei o filme inteiro esperando um "Bitch" do Pinkman, e infelizmente não tinha. Gostei do filme. O que *Velozes e Furiosos* tinha pego emprestado para a criação da franquia, *Need for Speed* pegou um pouco do *Velozes e Furiosos,* mas só um pouco. O filme acaba inovando nas tomadas de corrida, câmeras GoPro a vontade, dubles e menos efeitos visuais. As cenas do Cockpit dão uma imersão bem bacana, assim como as câmeras externas do lado do carro lembrando muito o jogo, assim como a presença dos policias de uma forma sem noção para parar os carros (que por sinal são fantásticos, algo que a franquia do Velozes e Furiosos não mostrou ainda) e os designs de mapas e outras coisas. A história e motivação que faz o personagem de Aaron Paul entrar na polêmica corrida do submundo Deleon não tem sentido nenhum, até a repercussão de uma das atitudes de um dos personagens acaba sendo meio blá, porem, o que vale realmente são as corridas e perseguições em takes e situações que lembram muito o jogo. A minha surpresa do filme também se deu pela participação de Michael Keaton como o criador da tal corrida do submundo.. rs...Pode ir sem medo, a diversão é bacana, certeza que você vai querer jogar. #AVANTE

Por: Alessandro "Bob"Bernard

Assista o trailer:
https://www.youtube.com/watch?v=fsrJWUVoXeM

ASSASSIN'S CREED IV: BLACK FLAG

Fonte: http://renegadoscast.com/wp-content/uploads/2014/01/Vitrine_Postx1_ASSASSINS.jpg

Este na verdade é o sexto jogo para consoles da série, sem contar com os DLCs de cada um e de versões para portáteis. E desta vez temos um assassino completamente novo porém com um sobrenome Familiar. Edward Kenway, que na cronologia assassina é o Avô do protagonista de Assassin's Creed III, Ratonhnhaké:ton conhecido carinhosamente por nós como Connor.

Fonte:http://renegadoscast.com/wp-content/uploads/2014/01/IMAG_Post_X1_ASSASSINs.jpg

Este jogo nos leva de volta ao Caribe do início do século 18, o berço dourado da pirataria, onde grandes nomes como Charles Vane, e Edward Teach, o temível Barba Negra, fizeram sua fama.

Conhecemos Edward, um marinheiro trabalhando com corsários, com uma esposa grávida em casa e sonhos de riqueza e de uma vida melhor que a de um empregado. A situação toda é mudada quando sua esposa o deixa, por não aguentar que seu marido não fique em casa e não se conforme com a vida que tem, fazendo assim com que Ed se entregue completamente ao trabalho corsário. Sua sorte muda quando o navio em que esta é afundado e acaba se tornando naufrago em uma ilha, onde se encontra com um dos caras que estava no outro navio. Curiosamente o cara veste um manto muito familiar com um capuz. Após matar o infeliz e roubar seu manto, começa a aventura de Edward Kenway e o jogo propriamente dito.

A primeira coisa em que fiquei atento foi para os lendários Bugs do Assassin's Creed. Quem jogou o III sabe do que estou falando. Não são tão frequentes, mas quando acontecem, realmente incomodam. Ficar preso entre a terra e o limbo, e outras coisas assim são bem chatas. E adivinhem só? Ainda estão lá! Que tristeza Ubisoft... mas isso é só um pequeno problema em meio ao um turbilhão de qualidades. Este, na minha opinião é o jogo mais divertido da série. O que menos cansa de jogar, pois além da trama principal, existem milhares de coisas para se fazer. E para um jogo como este, o fator "repetitivo" é quase que esperado, mas aqui não aconteceu. As missões em suma, seguem alguns padrões? Sim. Mas todas possuem elementos diferentes e possibilidades diferentes. Aqui temos as tradicionais missões de Siga e Mate, Escute a Conversa, Lute Com Uma Renca De Gente e Ache Aquele Artefato/Lugar. Mas fora isso há também as missões navais, feitas no comando de seu navio, o Jackdown ou Gralha, como preferirem. Missões essas que sentimos o gostinho no Assassin's Creed III e as vimos em toda a plenitude e diversão no IV. Só que desta vez você não é o bom moço que era o Connor. Você é um maldito pirata! Um mandrião! E como tal depois de derrotar outro navio, você pode (deve) saqueá-lo, no melhor estilo Piratas do Caribe, com marujos pulando para o outro navio, disparando suas armas e muita porradaria com espadas no convés inimigo. Na boa, é difícil cansar de fazer isso. Sem falar do cenário. Os mares Caribenhos transformados em um dos maiores cenários de mundo aberto que já tive o prazer de explorar a navio em um game.

Um dos pontos onde eu havia criado uma expectativa grande, foi o jogo fora do Ânimus. Ou seja, fora da pele do capitão Ed

e na pele de... seja lá quem for. Pois é. O fim do jogo anterior tornava difícil de imaginar como fariam uma sequência. E pra minha surpresa a Ubisoft, deu um jeito diferente, bem criativo e que abre a possibilidade de continuar a franquia tranquilamente por mais 10 jogos. A surpresa vem ainda antes da entrada do game. No anuncio dos produtores. Quem manja dos paranauês vai se ligar do que estou falando.

No fim das contas, a Ubisoft novamente acertou a fórmula para mais um jogo desta magnífica franquia da qual sou muito fã (apesar dos bugs. Por favor Ubisoft, de um jeito nisso!!) Com a fórmula que nós já conhecemos, misturada a novidades bem legais, uma trama legal e empolgante e um tema tão legal e que estava tão carente de algo bom na cultura pop. Altamente recomendado, mesmo se não jogou os títulos anteriores.

Por: Epic Eric

RYSE: SON OF ROME

Fonte:http://renegadoscast.com/wp-content/uploads/2013/12/Vitrine_X13_ryse.jpg

Depois de muita pesquisa, optei pela compra de um XBOX ONE, não só pelos jogos, mas pelo serviço de internet e distribuição digital que é impecável! E o comando de voz do Kinect 2.0 é de uma rapidez foda, e totalmente em português! Depois do belo presente/aquisição, resolvi começar pelo jogo que já tinha me chamado atenção logo de cara : RYSE | SON OF ROME, jogo que tinha muitas promessas e expectativas, pelo gráfico e história... e agora com o lançamento do console e jogo, será que vale mesmo?

SINOPSE

"Ryse: Son of Rome" conta a história de Marius Titus, um jovem soldado romano que testemunha o assassinato de sua família pelas mãos dos bárbaros, e então viaja com

Fonte:http://renegadoscast.com/wp-content/uploads/2013/12/Vitrine_X12_ryse.jpg

o exército romano para a Britânia buscando vingança. Marius vai evoluindo dentro do exército e quando menos espera, acaba tendo que se tornar um líder e defender o Império Romano (É engraçado a tamanha semelhança com "Gladiador" do Russell Crowe e Ridley Scott)

Fonte:http://renegadoscast.com/wp-content/uploads/2013/12/Vitrine_X11_ryse.jpg

REVIEW RENEGADO

Relamente, Jogo de Nova Geração! A Iluminação, gráficos e mil coisas, nos detalhes, traz a imersão garantida e integrante. Alguns dirão que o jogo é repetitivo pela sua jogabilidade, mas não tira o mérito do roteiro e da história. Nada que vai mudar a vida de alguém ou marcar como grandes jogos como GTA V, Shadow of Colossus e tal.

A inclusão de alguns comando de voz do novo kinect, deixam o jogo mais dinâmico e participativo com o jogador (cuidado com seus vizinhos, já que gritar "Disparar Descarga" parece que você está gritando no banheiro, hehe), tudo criado com a bela CryEngine 3! Claro que também tem algumas falhas, como a câmera que atrapalha um pouco ou a falha com uma punição de uns botões nas câmeras lentas.

E acho que é isso! Belo começo de nova geração para o XOne, tanto que estou ansioso para outros jogos como Battlefield 4 e Forza. Vale a pena muito o Videogame e seus jogos, sem contar voz de comandos para qualquer momento da jogatina e outras coisas! Acertaram na dose e agora é só curtir!!!

Por: Alessandro "Bob" Bernard

Season Premiere

BREAKING BAD

Fonte:http://renegadoscast.com/wp-content/uploads/2013/09/SP2.jpg

"Eu não estou em perigo, eu sou o perigo." – Walt

Demorei muito para começar a assistir Breaking Bad, não curti o primeiro episódio e achava muito enrolado. Não dava para entender muito bem o que era esse hype foda em uma série como essa. Depois de 4 tentativas (devo agradecer às aulas de direção da Sra. Bob), dei mais uma chance. Cara, foi como as tentativas do Jesse de ligar a van, e aí deu certo!

Tudo fazia sentido! Nunca esperava que uma série com esse tipo de trama pudesse me prender assim, de forma que eu consegui devorar todas as 4 primeiras temporadas e me diverti tanto quanto vocês nessa Quinta Temporada (claro, vocês que acompanham a mais tempo do que eu, rs). Foi 1 mês e meio de "Bitches", Foda pra CA$#@#$@, Meu Deus, e tudo mais palavras que não são faladas desde o tempo das construções das pirâmides do Egito.

Quanto a sinopse, vamos lá: Breaking Bad se passa em Albuquerque (Novo México, EUA). Encontramos um professor de química do ensino médio, chamado Walter White (Bryan Crasnton, Foda pra car@#$@#$), seu filho que sofre de paralisia cerebral, Walter Jr (RJ Mitte, sensacional) e sua esposa Skyler White (Anna Gunn), que está grávida. A merda no ventilador acontece quando Walt recebe a bela notícia que está com câncer de pulmão. Pensando no futuro da sua família, ele parte para a produção e venda de metanfetaminas (droga altamente viciante, com efeitos como a euforia, aumento do estado de alerta, da autoestima, do apetite sexual e pela intensificação de emoções) junto com seu ex-aluno Jesse Pinkman (Aaron Paul, Bitches!).

As consequências dessa parceria e as mudanças de comportamentos desses personagens são o que fascinam em todo o seriado. A riqueza dos detalhes, das pontas, toda causa e reação são cuidadosamente tratados de uma forma peculiar espetacular, que te prende e faz você querer ver mais e mais, sobre até onde um cara pode se arriscar para garantir o bem da sua família.

Tamanho o cuidado em detalhes, Breaking Bad só vem conquistado cada vez mais prêmios, nomeações e fãs (Acabou de entrar para o Guinness Book como a série mais bem avaliada de todos os tempos). Tamanho cuidado que cada intro dos episódios tem uma sacada foda, até mesmo nos créditos iniciais. Para aqueles que não sabem, os símbolos de elementos químicos da Tabela Periódica em verde nos nomes dos atores, a fórmula $C_{10}H_{15}N$, que é a própria fórmula molecular da metanfetamina (10 átomos de carbono, 15 átomos de hidrogênio e um átomo de nitrogênio) e sua massa molecular , 149.24. Se isso é só na abertura, imagine em toda a série!

E como todo fã de Breaking Bad, eu vou falar também para aqueles que não fazem ideia do que é: ASSISTAM!!! É MUITO BOM!!! Vou deixar o link aqui para você adquirir as séries (caso for um viciado colecionador igual a mim!). Mas se você não pode no momento (ou não quer, rs) aproveitem no Netflix, que possui as 5 temporadas (a quinta está até o 8º episódio) e ainda nesse mês passará o Episódio Final , o tão aguardado episódio 16 , que vai ao ar dia 29 de Setembro.

Por: Alessandro "Bob" Bernard

BATES MOTEL

Fonte: http://renegadoscast.com/wp-content/uploads/2013/11/SP5.jpg

"Mãe, você é tudo. Tudo para mim...E eu não quero viver no mundo sem você!"- Norman Bates

Acho que todos aqui conhecem Psicose, um desses filmes clássicos mais aclamados do gênio Hitchcock (Se você ainda não viu, por favor, por mais que seja velho, vale a pena ver o original!), e o que essa série nos traz é muito mais do que um simples prólogo.

Aqui você entende a origem da personalidade de Norman Bates (Freddie Highmore, que não é mais aquele garotinho simples que foi visto no Remake da Fantástica Fábrica de Chocolate), um garoto de 16 anos, que tem aquele apego bem "afetivo" com a sua mãe, Norma Louise Bates (Vera Farmiga, sensacional atriz). Após a morte do Pai de Norman, os dois resolvem começar a vida em outra cidade, Oregon, e resolvem comprar um Motel de beira de estrada em conjunto com um casarão bem conhecido pelos fãs do filme.

A partir desse momento e da mudança, a luta da mãe para proteger o seu filhote, vai traçando a personalidade de Norman, desde a possessividade da Norma com o Norman, a volta do meio irmão de Norman, Dylan Massett (Max Thieriot) que acaba agregando mais a trama, assim como outros personagens no qual é bacana a naturalidade do surgimento deles e o que cada ação na série tem a sua devida consequência. É bom ressaltar também que todo esse ambiente foi trazido para os tempos atuais. Então não se assustem se acabar vendo smartphones a rodo por ai.

Uma coisa que também é interessante são as peças publicitárias, no qual pegaram alguns detalhes mórbidos da série que tem a ver totalmente com o drama. O legal é ver como todas elas tem sua ligação com a série e a cada coisa que vai surgindo você compara com os posters e acaba soltando aquela coisas, tipo " Que Fod!@!@". A série já acabou sua primeira temporada e a segunda já esta confirmada. Podem curtir a vontade, são 10 capítulos de 45 min em média, produzido pela Universal Television e exibido pelo Universal Channel por aqui (se não já acabou também, rs).

Aproveitem que já terminou a primeira temporada e ASSISTAM!

Por: Alessandro "Bob" Bernard

ALMOST HUMAN

Fonte: http://renegadoscast.com/wp-content/uploads/2014/02/SP_VITRINEPOST_ALMOSTHUMAN.jpg

"O DRN é bom para você" – Capitã Maldonado

Estamos no ano de 2048, a evolução da tecnologia começou a fugir do controle e a polícia não está preparada para os criminosos com esse tipo de tecnologia em mãos, por isso cada policial humano deve ter como parceiro um "sintético", mas, assim como o Dorian, eu não sou muito fã desse termo.

A história gira em torno do Detetive John Kennex, que, depois de uma emboscada, perdeu uma das pernas e teve que substituí-la por uma perna sintética, o que o deixou um tanto quanto estressadinho ao voltar ao trabalho.

Fonte: http://renegadoscast.com/wp-content/uploads/2014/02/SP_VITRINEPOST_ALMOSTHUMAN.jpg

Depois de um pequeno assassinato acidente com o MX-43 – androides feitos para trabalhar racionalmente, a partir de cálculos e todas as coisas chatas racionais – designado a ele como parceiro, ele é colocado com o DRN, Dorian – os DRN são feitos a partir de um programa chamado Alma Sintética, que faz com que eles tenham, como todo humano, intuição, emoções e, portanto, dependendo do por quanto eles passam, isso pode fazer com que não estejam mais aptos para o trabalho, então esse tipo de androide foi desativado devido a muitos defeitos e, assim, substituídos pelos MX.

O que me chamou a atenção na série foi a relação de amizade entre Kennex e Dorian, uma amizade tão improvável em que o robô humaniza o humano. Além disso, as investigações policiais são bem variadas e bem interessantes.

Por: Bea

JOGADOR NÚMERO 1

Fonte:http://renegadoscast.com/wp-content/uploads/2013/12/Vitrine_X13_ryse.jpg

SINOPSE

Cinco estranhos e uma coisa em comum: a caça ao tesouro.

Achar as pistas nesta guerra definirá o destino da humanidade. Em um futuro não muito distante, as pessoas abriram mão da vida real para viver em uma plataforma chamada OASIS. Neste mundo distópico, pistas são deixadas pelo criador do programa e quem achá-las herdará toda a sua fortuna. Como a maior parte da humanidade, o jovem Wade Watts escapa de sua miséria através do OASIS. Mas ter achado a primeira pista para o tesouro deixou sua vida bastante complicada. De repente, parece que o mundo inteiro acompanha seus passos, e outros competidores se juntam à caçada. Só ele sabe onde encontrar as outras pistas: filmes, séries e músicas de uma época que o mundo era um bom lugar para viver. Para Wade, o que resta é vencer – pois esta é a única chance de sobrevivência.

CRÍTICA RENEGADA

Antes de começar a escrever qualquer coisa sobre esse livro, preciso dizer que é a melhor obra que li em anos. O Autor Ernest Cline conseguiu sintetizar em um único livro tudo que gosto e admiro. Tecnologia, conflito e Ação. E nesse caso ainda ganhamos um Bônus: Anos 80.

O Livro é uma imensa homenagem a cultura pop dos anos 80. Filmes, seriados, música e principalmente jogos. Você que viveu essa época colado na televisão e jogando o seu ATARI ficará maluco com todas as referências que ele faz.

A história do livro se baseia na vida de James Halliday, o criador da plataforma OASIS que pode ser considerado um "Second Life" que deu muito certo. A vida real se mistura com a virtual através desse sistema, a ponto de que o dinheiro que você tem dentro do OASIS valha para a vida real também.

Quando Halliday morre ele deixa como testamento toda sua fortuna, avaliada em mais de 200 bilhões, para o jogador que conseguir resgatar as três chaves e abrir os três portões que ele deixou espalhado pelo OASIS e assim se tornar "O Jogador número 1".

No meio disso tudo está o personagem principal: Waden Watts, um jovem pobre que dedica sua vida a desvendar os mistérios deixados por Halliday. Sua vida muda quando ele se torna o primeiro jogador a abrir o primeiro portão e ser caçado dentro e fora do OASIS.

Mas vou parar de escrever sobre o enredo e dizer o porque do livro ser tão especial. Simplesmente você começa a fazer parte da história e vive página a página o drama de Waden. Em certo momento do livro você acaba se confundindo sobre o que é real e sobre o que é virtual. Tudo se mistura entre caçadas, quebra cabeças, descobertas envoltos em um ambiente dos anos 80. Romance? Lógico que sim. Mas sem ser chato e seguindo fielmente a temática do livro.

Se eu pudesse escolher apenas um livro para ler em 2014, com certeza eu não pensaria duas vezes: Jogador Número 1.

Por: Marco Aurélio Dias

MAZE RUNNER - CORRER OU MORRER

Fonte:http://renegadoscast.com/wp-content/uploads/2013/12/ERMZ1.jpg

SINOPSE

Ao acordar dentro de um escuro elevador em movimento, a única coisa que Thomas consegue lembrar é de seu nome. Sua memória está completamente apagada. Mas ele não está sozinho. Quando a caixa metálica chega a seu destino e as portas se abrem, Thomas se vê rodeado por garotos que o acolhem e o apresentam à Clareira, um espaço aberto cercado por muros gigantescos. Assim como Thomas, nenhum deles sabe como foi parar ali, nem por quê. Sabem apenas que todas as manhãs as portas de pedra do Labirinto que os cercam se abrem, e, à noite, se fecham. E que a cada trinta dias um novo garoto é entregue pelo elevador. Porém, um fato altera de forma radical a rotina do lugar – chega uma garota, a primeira enviada à Clareira. E mais surpreendente ainda é a mensagem que ela traz consigo. Thomas será mais importante do que imagina, mas para isso terá de descobrir os sombrios segredos guardados em sua mente e correr, correr muito.

CRÍTICA RENEGADA

Plong. Mértila. Trolho. Clareano. Aguadeiro. Socorrista. Verdugo.

Sei que essas palavras são estranhas, mas ao terminar de ler o primeiro livro da série Maze Runner elas serão comuns em seu dicionário. O Autor James Dashner sem nenhuma explicação prévia as coloca em sua estória e o mais impressionante é que ao decorrer do livro elas fazem sentido.

Comecei lendo Maze Runner com uma expectativa muito grande. Foram várias indicações de amigos e o anúncio do filme para o ano de 2014. Tenho que ser honesto e dizer que mesmo sendo um livro infanto juvenil, não é fácil de se ler. Como eu comentei logo acima, são várias palavras estranhas colocadas em um ambiente totalmente novo e confuso, e você demora um tempo para se adaptar a tudo isso.

Mas eu aconselho, aconselho não, peço encarecidamento que vocês passem por esse começo confuso e continuem lendo porque vale muito a pena. Parece que de uma hora para outra você começa a entender tudo. Aquele trecho la no começo que não fez nenhum sentido quando você leu, você acaba descobrindo que ele faz todo sentido do mundo.

Dito isso, eu gostaria de falar do personagem principal, Thomas. É pelos olhos dele que a estória é contada e são os sentimentos dele que você acaba sentindo. Eu adorei o Thomas. Desde o começo o autor consegue passar a insegurança, suas dúvidas, todas as suas emoções de uma forma bem clara, inclusive a obsessão inexplicável dele de sair do labirinto. Por mais que ele seja o personagem principal e toda a trama seja baseada em suas ações, você sente medo por ele. Não há certeza de que ele chegará ao final e na minha opinião isso é fenomenal.

Tenho duas pequenas críticas ao livro: A primeira é o começo que chega a ser chato, mas como disse antes vale a pena insistir e acreditar na obra. A Segunda são alguns personagens importantes que são colocados de forma repentina ou forçada pois ele é essencial para o desenvolvimento. Mas deixo bem claro que nenhuma dessas críticas diminue o valor da obra ou sua intensidade.

A minha maior preocupação agora é em relação ao filme. Pela complexidade e detalhamento do livro creio que a adaptação não será simples. Mas com certeza estarei no cinema no dia do lançamento. Afinal... EU QUERO VER UM VERDUGO!

Por: Mike

1984

Fonte: http://renegadoscast.com/wp-content/uploads/2014/02/Vitrine_1984.jpg

SINOPSE

Winston vive em uma sociedade completamente dominada pelo Estado, onde tudo é feito coletivamente, mas cada qual vive sozinho. Ninguém escapa à vigilância do Grande Irmão, a personificação de um poder cínico e cruel ao infinito. De fato, a ideologia do Partido dominante não visa nada de coisa alguma para ninguém, no presente ou no futuro. Para o partido só interessa o poder em si. Nem riqueza, nem luxo, nem vida longa, nem felicidade – só o poder a qualquer custo.

CRÍTICA RENEGADA

O livro é considerado, por muitos, a obra literária mais importante do século XX e a melhor distopia já publicada.

O mundo que conhecemos está dividido em três continentes: Oceânia, Eurásia e Lestásia. O personagem principal, Winston, mora na Oceânia e faz parte do Partido do Grande irmão, que está atualmente no poder. Tudo e todos são controlados pelo partido. Desde seus atos, seu linguajar a até mesmo seus pensamentos. Winston pertence a uma pequena parte da população que trabalha diretamente para o partido e colabora para que ele continue sempre no poder.

Dentro da casa de todos existe uma "Teletela", um tipo de televisor que envia imagens e observa tudo que acontece dentro de sua própria casa. No topo do poder está o grande irmão, o rosto estampado em todos os anuncios do Partido. É ele quem incentiva e impulsiona os aliados a fazerem apenas o que o partido deseja.

Uma das principais funções do partido é reescrever o passado para que dessa forma consiga controlar o presente e, consequentemente, o futuro.

Pensar, ou melhor, duplipensar que nada mais é manter duas crenças contraditórias na mente ao mesmo tempo, é considerado o crime mais grave para o partido fazendo com que a Polícia das idéias entre em contato imediatamente com o acusado e o mesmo nunca mais seja visto.

Depois dessa introdução eu preciso dizer que esse livro é o último romance escrito por George Orwell e tentar explicar a todos o impacto que essa história teve e tem na humanidade até os dias de hoje. O autor deixou como testameno um livro mostrando como um sistema totálitário pode ir aos poucos tomando conta da sua vida, e quando você menos perceber você já faz parte do processo. 1984 não é um livro fácil de se ler, em alguns momentos a escrita é até mesmo meio arrastada, posso dizer até mesmo chata, mas seu valor é inigualavel.

Tudo no livro é interessante, desde a vida dos personagens, suas atitudes e as pequenas coisas que acontecem em seu dia a dia. O dois minutos de ódio, as atribuições de cada função dentro do partido, a forma como tratar o sexo oposto e o mecanismo que sempre gira para que quem está no poder, continue sempre lá.

O livro fala de privações que temos em nosso dia a dia e não damos valor, por exemplo é proibido escrever, pois escrevendo você começa a ter idéias. Você não poder namorar, não pode comer ou beber nada que não seja fornecido pelo partido. Em 1984 você aprende o valor da liberdade. Tanto que uma das frases mais marcantes do livro é: "Liberdade é poder dizer que 2 + 2 = 4".

Essa obra influenciou filmes, livros, propaganda e até mesmo a televisão. Não podemos esquecer que Reality shows são inspirados na premissa de você poder assistir tudo que se passa em uma casa, um desses programas inclusive carrega o nome do principal personagem do partido: Big Brother.

Uma das propagandas mais emblemáticas já produzidas foi feita pela Apple em 1984 usando como plano de fundo todo o ambiente do livro e comparando a IBM com o grande irmão. Foi passada apenas

uma vez durante o Super Bowl e até hoje é comentada por profissionais, inclusive foi dirigida por Ridley Scott que na época tinha acabado de dirigir Blade Runner.

O FILME

Foi produzido exatamente em 1984 tendo John Hurt no papel principal. Na minha opinião conseguiu chegar bem próximo ao ambiente e narrativa do livro. Quando eu li o livro tudo parecia sem graça, a comida parecia sem sabor, podemos dizer que tudo era "Cinza". Quando eu vi o filme tudo pareceu exatamente como eu li, tudo feito de forma mecânica. Esse é um dos filmes que você pode ler o livro antes, porque ele respeita bastante a obra original. Teve outro filme feito nos anos 50 mas sinceramente eu não assisti.

CURIOSIDADES:

- As vendas do livro subiram 7000% quando os escandalos de espionagem dos estados unidos vieram a tona.

- Em 2011 um senador dos Estados Unidos usou como base o livro 1984 para derrubar um projeto de Monitoramento de indivíduos.

- David Bowie já lançou uma música inspirada no livro. O título? 1984.

- V de vingança foi profundamente inspirada em 1984. Inclusive no filme, John Hurt está no elenco também.

- Em 2011 a Microsoft criou um programa que captava as telas do usuário sem que ele soubesse. Foi apelidado de 1984 com toda razão.

- O espólio de George Orwell processou a CBS pela criação do programa Big Brother.

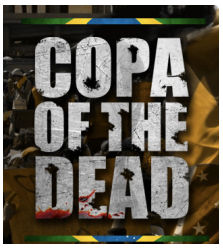
Sendo assim, encerro essa coluna que está enorme pedindo a todos que leiam essa obra prima, vejam o filme e se interessem mais pelas de George Orwell.

E não se esqueçam: "O Grande irmão zela por ti".

Por: Mike

APPs Renegados

COPA OF THE DEATH



Se o Apocalipse Zumbi acontecesse durante a copa do mundo, que tipo de sobrevivente você seria?

Acesse e descubra: https://www.facebook.com/Renegados.RC/app_208195102528120

H2HC

HACKERS TO HACKERS CONFERENCE

MAGAZINE

ANUNCIE NA H2HC MAGAZINE!

Sua marca na revista do mais antigo evento de pesquisas em segurança da informação da América Latina!

entre em contato conosco!

revista@h2hc.com.br

Horóscopo

Áries

O Sol começa a caminhar através de Peixes e inicia-se uma fase bem bugada. O momento pede maior cuidado, você anda com muitas vulnerabilidades, seu anti-virus não anda servindo para muita coisa.

Touro

A sua vida está conturbada, você fez muita POG no passado, agora vai pagar o preço.

Gêmeos

A sua fase não anda muito boa, faça mais testes em seus programas. Grandes riscos de seu compuntador entrar em uma Botnet, é bom começar a se preocupar.

Câncer

O Sol começa a caminhar através de Peixes e seu foco passa a ser seus projetos futuros, especialmente os que envolvem pessoas e empresas estrangeiras. Está na hora de você rever o seu fuzzer.

Leão

Você está em uma excelente fase, graças a sua engenharia social várias portas novas se abriram, aproveite.

Virgem

Sua vida profissional anda um pouco conturbada, mas nao é culpa sua. Seja esperto ative um IDS em seu sistema e verá o que realmente esta acontecendo. No amor grandes chances de encontrar o Zero Day da sua vida.

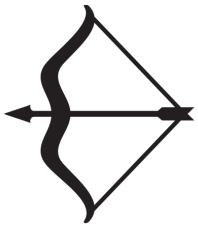
Libra

Momento de muita interferencia na sua vida. Você precisa parar de deixar as crianças brincarem no seu computador, elas não sabem onde clicam.

Escorpião

Melhore seu ânimo, a fase ruim já está passando, mas para isso sua contribuição é necessaria. Pare de agir como um .gif chato e brilhante.

Sagitário



Existem momentos que formatar o disco é a melhor saida. Arrisque.

Capricórnio

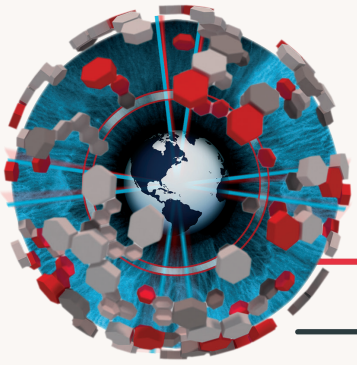
Excelente fase para inovar, ouse, escolha novas linguagens e comece a programar.

Aquário

Você está pior que um Keylogger, pare de registrar tudo, sua paranoia não vai te levar a nada.

Peixes

Instale um Rootkit em sua vida. As pessoas não precisam saber tudo que você faz.



H2HC

HACKERS TO HACKERS CONFERENCE

MAGAZINE