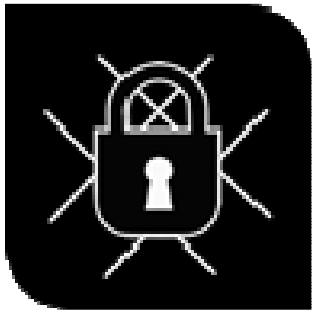


HTExploit

Bypassing .htaccess and beyond!

Matias Katz - @matiaskatz

Maxi Soler - @maxisoler

mkit | IT & Security Solutions

www.mkit.com.ar

Matías Katz (@matiaskatz) is a Penetration Tester who specializes in Web security analysis. He loves to build simple tools to perform discovery and exploitation on any software or network. He is the founder and CEO of **Mkit Argentina**, a company that specializes in penetration testing and code auditing services.

Maximiliano Soler (@maxisoler) lives in Buenos Aires, Argentina and currently works as a Security Analyst. Maxi has discovered vulnerabilities in different applications Web and Microsoft's products.

.htaccess - What is it and what is it for?

`.htaccess = hypertext access`

It is a distributed ***configuration file*** that allows each directory and subdirectory to have its own configuration, without the need of reconfiguring Apache's main settings file.

.htaccess usually uses the same syntax as the Web server's main configuration files.

Some usage examples

Redirection	URL Redirection
Directory listing	URL Rewriting
Client-Server Dialogs	Personalized Error Messages

...

Authentication **Authorization**

Why attacking the protected directories?

Because it is common to find...

- **x** Backup files
- **x** Configurations
- **x** Outdated versions
- **x** New developments
- **x** Admin Logins **;)**

HTExploit (**H**iper**T**ext access **Exploit**)

It is an open-source tool written in Python that exploits a **weakness** in the way that .htaccess files can be configured to protect a web directory with an authentication process.

By using this tool you will be able to list the contents of a directory protected this way, **bypassing the authentication process**.

The tool was presented at **Black Hat USA 2012 Conference**

- Free and Open Source
- User-friendly
- Flexible
- Modularized
- Reporting
- Integrated with other major tools
- Multiplatform

- It is an old weakness, not currently exploited by other tools
- Most websites recommend to create vulnerable .htaccess configs
- We could not find any other tool that met our needs
- Research for fun and profit!

- x Not a new vulnerability
- x Not a 0-day
- x Not a one-click Pwnage tool
- x Not a replacement for other web hacking tools

`AuthUserFile /[FOLDER]/.htpasswd`	Full path to the htpasswd file
`AuthName "Protected Area"`	Login screen title message
`AuthType Basic`	Required line
`<Limit GET POST>`	Initiates the GET and POST methods limit
`require valid-user`	Sets access restriction for a valid user only
`</Limit>`	Ends the limit tag

DEMO

- The problem resides in how HTTP requests are being limited.
- The following statement:

```
<Limit GET POST>
    require valid-user
</Limit>
```

Indicates that the "require valid-user" directive only applies to GET and POST methods. Therefore, any other non-standard HTTP request would be **ALLOWED** and allows the request without requiring authentication

- When PHP receives the non-standard method, it processes it like a GET, downloading the requested file.

- Improve the .htaccess configuration
- Apache module (experimental)
- PHP Code

`AuthUserFile /[FOLDER]/.htpasswd`	Full path to the htpasswd file.
`AuthName "Protected Area"`	Login screen title message.
`AuthType Basic`	Required line.
`<Limit GET POST>`	Initiates the GET and POST methods limit.
`require valid-user`	Sets access restriction for a valid user only.
`</Limit>`	Ends the limit tag.

`<LimitExcept GET POST>`
`Order Allow,Deny`
`Deny from all`
`</LimitExcept>`
	Restricts access control to all HTTP methods except the ones **specified** above.

Apache Module mod_allowmethods

Easily restrict what HTTP methods can be used on the server.

Status: Experimental | **ID:** allowmethods_module | **Source:** mod_allowmethods.c

```
<Location />
   AllowMethods GET POST OPTIONS
</Location>
```

Notes:

- The HTTP-methods are case sensitive, and are generally as per RFC given in upper case.
- mod_allowmethods was written to replace the rather kludgy implementation of Limit and LimitExcept.

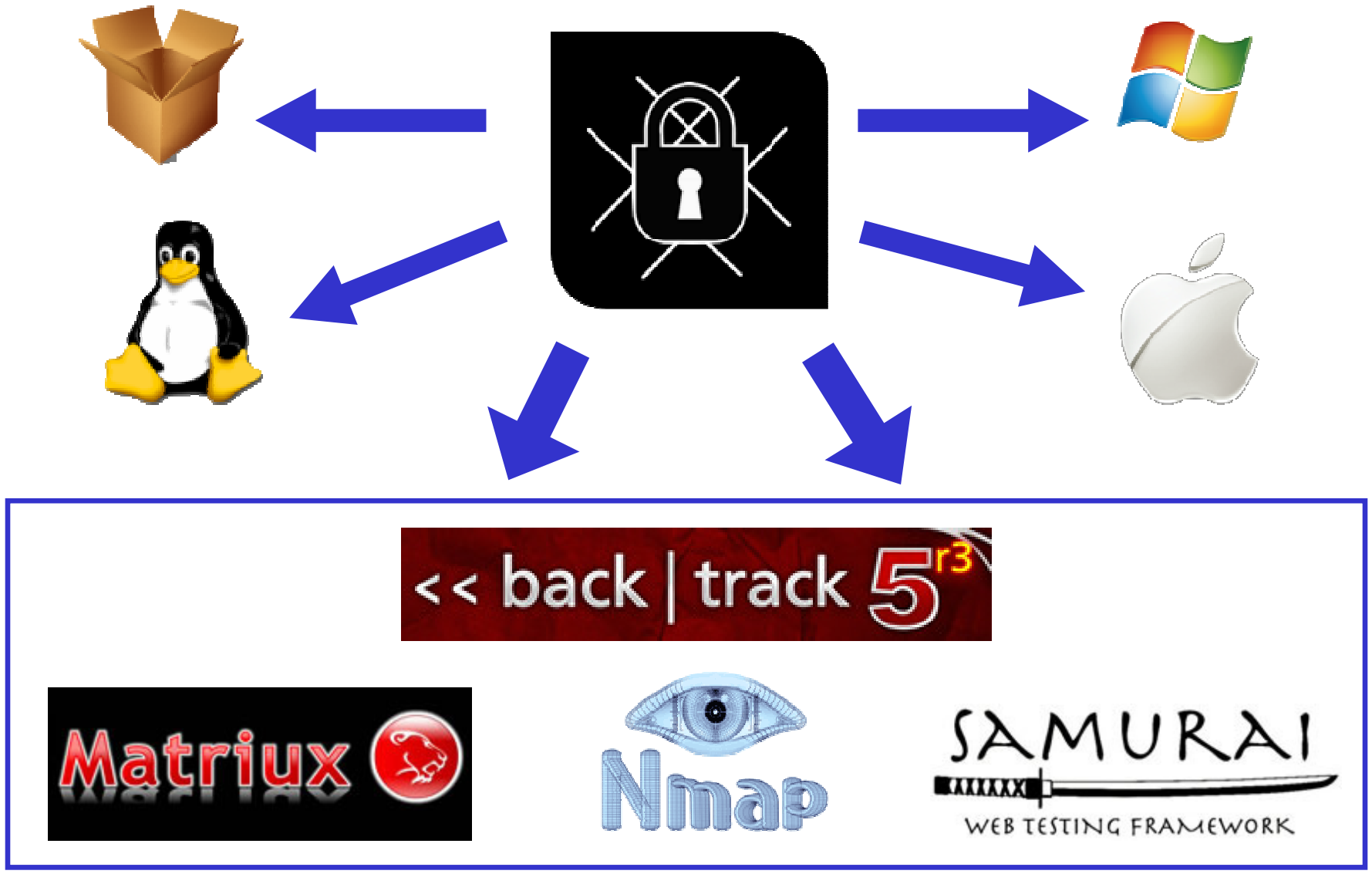
From the PHP code we could implement the following:

- **x** Check if the $PHP_AUTH_USER variable is set.

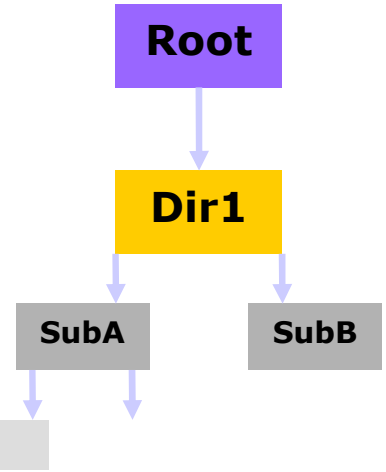
- **x** Check if $_SERVER["REQUEST_METHOD"] uses GET or POST, otherwise throw an error message.

- It is not enough to just declare the traditional HTTP methods. It is also necessary to restrict access to those unknown or unwanted.

- From the developer's perspective, it is mandatory to perform the necessary security checks, to be able to rely on more than just the configuration files.



Next Release v1.0



The latest version includes:

- ***Link Scanner*** - Crawling the website recursively
  - Remote File Inclusion
  - Local File Inclusion
  - SQL Injection

- **Integration With:**

metasploit®

Nmap

DirBuster

Questions

```
.d8888b.
d88P  Y88b
     .d88P
   .d88P"
   888"
   888

   888
```

- **HTTP Authentication: Basic and Digest Access Authentication**

http://tools.ietf.org/html/rfc2617

- **Apache Tutorial: .htaccess files**

http://httpd.apache.org/docs/2.0/howto/htaccess.html

- **Common Configuration Problems: Issue #81 (090597)**

http://www.apacheweek.com/issues/97-09-05#configerrors

- **Authentication, Authorization and Access Control**

http://httpd.apache.org/docs/2.4/howto/auth.html

g0t HTExploit?

www.htexploit.org

Thank you!!

Mkit | IT & Security Solutions

www.mkit.com.ar

Matias Katz

Twitter: @matiaskatz

Maximiliano Soler

Twitter: @maxisoler

The potential of any tool or technique is limited only by the ***imagination*** of the user.