Appsec & Infrastructure 101

$ whoami

[+] Nishaanth Guna, Application Security Lead, Appknox

[+] 5+ years in application and infra security

[+] Part-time bug bounty hunter

[+] Connect with me : @nishaanthguna at LinkedIn

Agenda

01 Request Smuggling into an internal network to access backend portals and poison legitimate user's request

02 External Infrastructure Pentest - Getting shell with UDF exploitation

03 Where and how of learning core fundamentals for application security

Request Smuggling?!

What is HTTP Desync attacks? Why is it caused?
What is the impact of this attack?

How to exploit? Can we affect other users?
Can we attack internal systems? Is it an RCE? Is it 0-click?

Demo on CL.TE Request Smuggling

Press Release

Read about a partnership with a new IoT vendor.

DNS Brute

Research the company, find the domain records and add it to recon list. Brute the company DNS

Port Scan IP

Port scan the IP and get a new CMS portal in a weird port

0-day in CMS

Find multiple zero days in the CMS and work on lateral movement and post exploitation

Spray and Pray

Spray passwords which you got, get into MySQL server

Shell Upload

Upload a shell to the server using mysql UDF exploitation

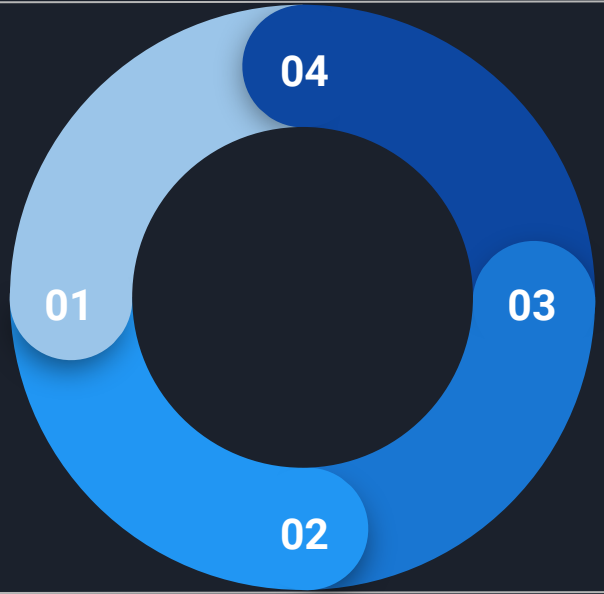
Sources to learn application security

Source Code Audits

Focus on one language. Review existing exploits, new 0-days. Sample exercise and why?

Conference Talks

Blackhat, Defcon, AppSec, PHDays



Strong basics

Networking, Programming, Security Concepts. What is SOP?

Move away from basics

SSRF, XXE, Cache Poisoning, XS-Leaks

Spot the vulnerability

19 lines (16 sloc) | 511 Bytes

```php
<?php

if( isset( $_POST[ 'Upload' ] ) ) {
        // Where are we going to be writing to?
        $target_path  = DVWA_WEB_PAGE_TO_ROOT . "hackable/uploads/";
        $target_path .= basename( $_FILES[ 'uploaded' ][ 'name' ] );

        // Can we move the file to the upload folder?
        if( !move_uploaded_file( $_FILES[ 'uploaded' ][ 'tmp_name' ], $target_path ) ) {
                // No
                $html .= '<pre>Your image was not uploaded.</pre>';
        }
        else {
                // Yes!
                $html .= "<pre>{$target_path} succesfully uploaded!</pre>";
        }
}

?>
```

Spot the vulnerability

9 lines (7 sloc) | 145 Bytes

```php
<?php

$html = "";

if ($_SERVER['REQUEST_METHOD'] == "POST") {
        $cookie_value = time();
        setcookie("dvwaSession", $cookie_value);
}
?>
```

Spot the vulnerability

30 lines (24 sloc) | 630 Bytes

```php
<?php

if( isset( $_POST[ 'Submit' ]  ) ) {
        // Get input
        $target = $_REQUEST[ 'ip' ];

        // Set blacklist
        $substitutions = array(
                '&&' => '',
                ';'  => '',
        );

        // Remove any of the charactars in the array (blacklist).
        $target = str_replace( array_keys( $substitutions ), $substitutions, $target );

        // Determine OS and execute the ping command.
        if( stristr( php_uname( 's' ), 'Windows NT' ) ) {
                // Windows
                $cmd = shell_exec( 'ping  ' . $target );
        }
        else {
                // *nix
                $cmd = shell_exec( 'ping  -c 4 ' . $target );
        }

        // Feedback for the end user
        $html .= "<pre>{$cmd}</pre>";
}

?>
```

Thank you!

Reference Links:

[1] https://portswigger.net/web-security/request-smuggling

[2] https://www.redteamsecure.com/blog/penetration-testing-vs-red-teaming