Offensive Approach to Hunt Bugs

Command Injection

Background Concept about CMDi

Operating system command injection vulnerabilities arise when an application incorporates user-controllable data into a command that is processed by a shell command interpreter. If the user data is not strictly validated, an attacker can use shell metacharacters to modify the command that is executed, and inject arbitrary further commands that will be executed by the server.

Impact of CMDi

By exploiting a command injection vulnerability an attacker can abuse the function to inject his own operating system commands. This means he can easily take complete control over a web server

Example of CMDi

A common function exists that passes an IP address the user specifies to the system's ping command. Therefore if the user specifies 127.0.0.1 as an IP address, the command will look like this:

```
ping -c 5 127.0.0.1
```

- Since it is possible to break out of the ping command or provoke an error with useful information the attacker can use this functionality to execute his own commands. An example for adding a second system command could look like this:
- `ping -c 5 127.0.0.1; id`

Possible Related Parameters

daemon host

upload dir execute download

log ip cli cmd

How to find There is a CMDi

- Use Delimiter to Break or Continue the execution of CMDs there
- Delimiter List

```
; ^ &

&&

|                  ||

%0D         %0A   \n

<
```

CMDi Hunting Example

Steps

- Find a Input Filed whose interacting with operating system shell
- Try to execute and system shell commands with delimiter

Example - ;ls

&&ls

||ls

CMDi Hunting Example

Live Web -

- http://projects.knmi.nl/

Exploitation of CMDi

1. Download Commix

https://github.com/commixproject/commix.git

Syntax

python commix.py -u URL

Example – python commix.py -u projects.knmi.nl/rico/RICO/archive/model/simfiledump.php?simfilename=