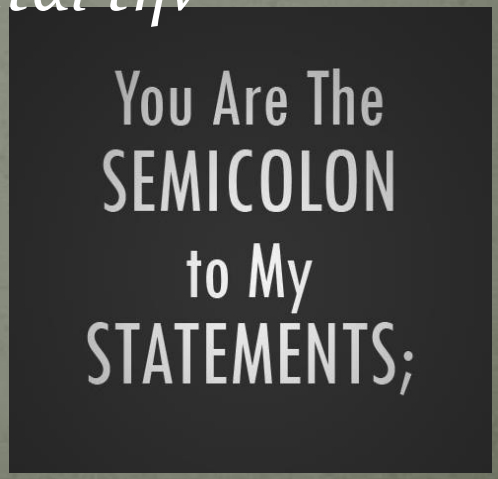
LFI & Code Execution

©Ανδρέας Βενιέρης 24/01/2016

Μπορούμε άραγε να οδηγηθούμε σε εκτέλεση κώδικα αφού βρούμε κάποιο LFI (Local File Inclusion) bug σε μια ιστοσελίδα;

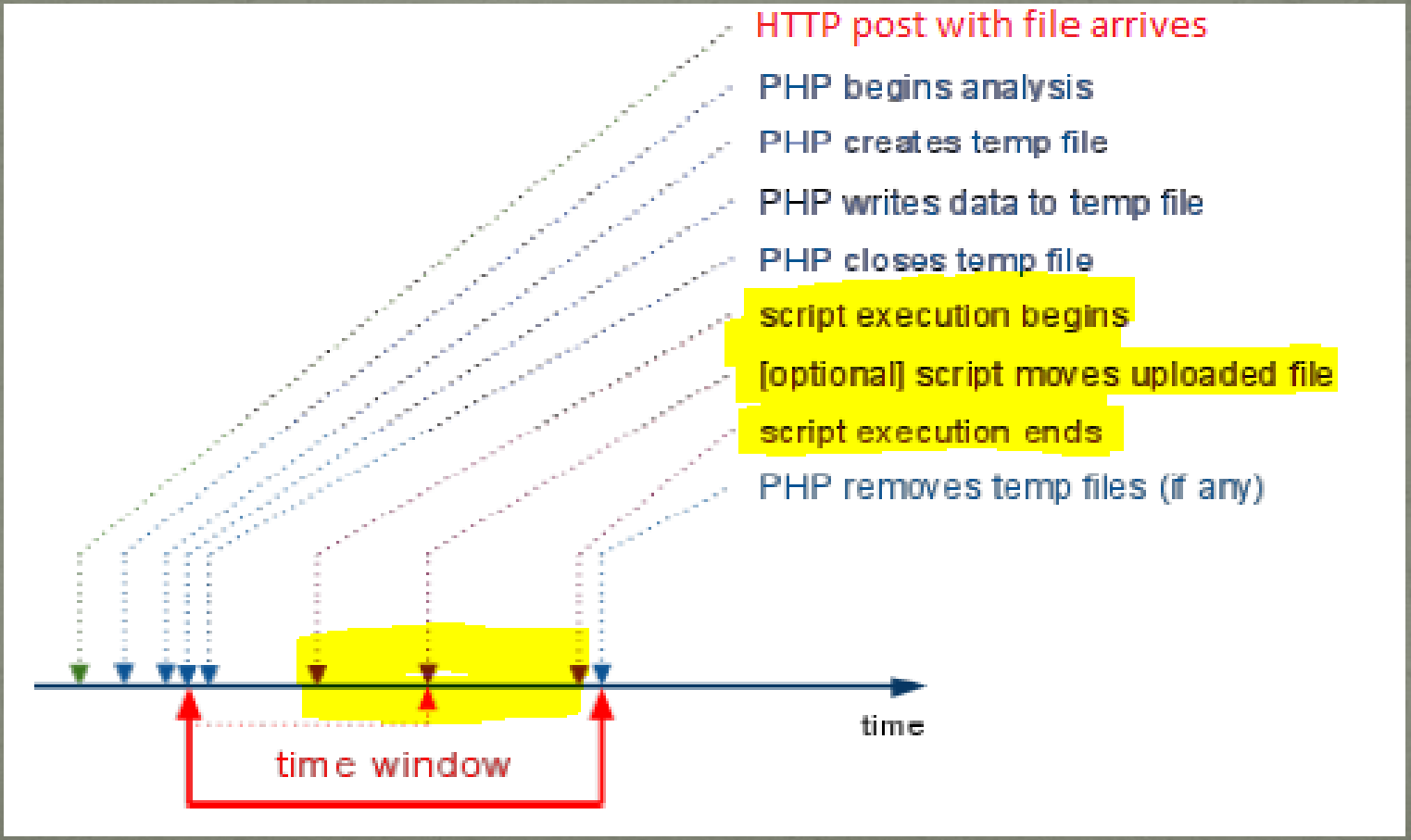
C:\> Για να δούμε τι θα δούμε..._

- *White Paper: LFI WITH PHPINFO() ASSISTANCE Insomnia Security, 6/09/2011.*
- *Ποια είναι η αδυναμία ή αλλιώς (...bug or a feature?).*
- *Κάτω από ποιες προϋποθέσεις μπορεί κάποιος να την εκμεταλλευτεί.*
- *Παρουσίαση του Xploit που εκμεταλλεύεται την παραπάνω αδυναμία.*
- *Μα υπάρχει ακόμα;;; (examples & stats)*



C:\> Ποια είναι αυτή η αδυναμία επιτέλους;

Η αδυναμία βρίσκεται στον τρόπο που λειτουργεί η function της PHP: *move_uploaded_file();*



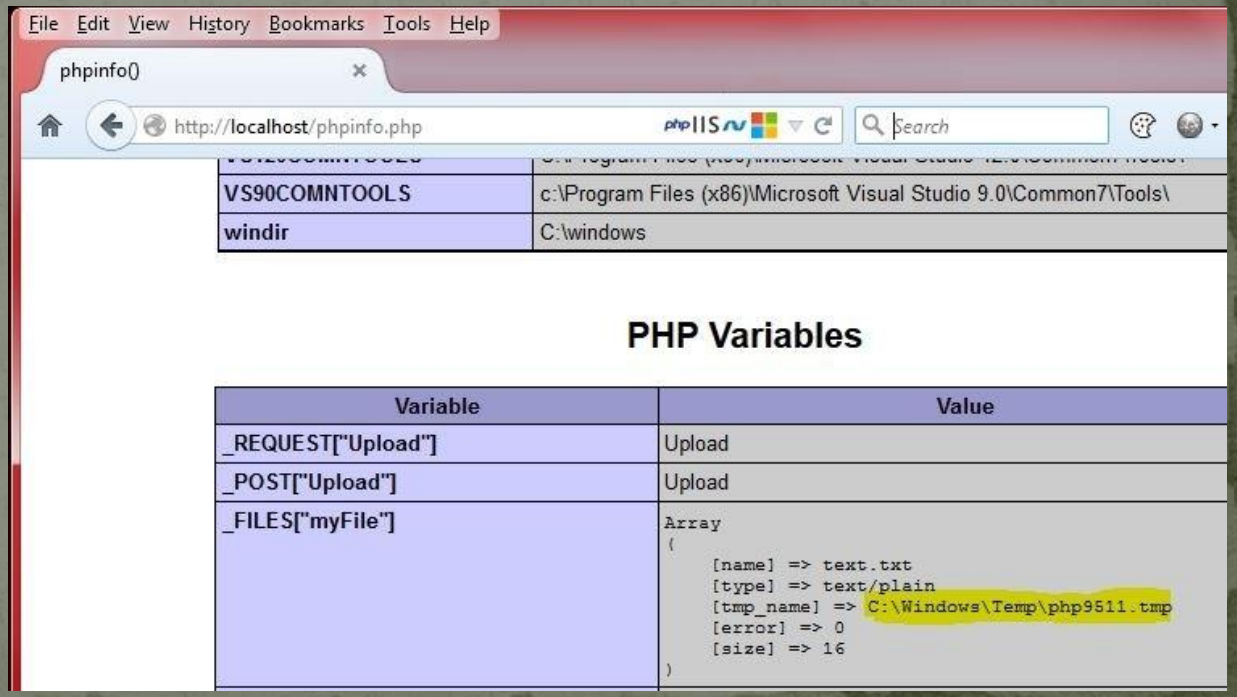
C:\> How To

1. «Τσιμπάμε» το όνομα του temp file.

1.1: MSWINDOWS *FindFirstFile()* bug (2007)!!
«**OPENFILE myfile<<** » → «**OPENFILE myfile*** » →*findfirstfile()* (Win API)

1.2: *Multi*thread *attack… no bug based ;)*
Στόχος: να καθυστερήσει τον Web Server μέχρι να καταφέρει κάποιο thread να πάρει το όνομα του προσωρινού αρχείο.

Phpinfo() Required!



C:\> Ok, we get the temp, now what...?

- Αφού καταφέραμε να πάρουμε το όνομα του προσωρινού αρχείου έχουμε μια μικρή (αλλά υπαρκτή) πιθανότητα, η επόμενη εντολή του exploit μας να βρει αυτό το προσωρινό αρχείο ακόμα στην θέση του (δηλαδή, να μην το έχει σβήσει ακόμα ο preprocessor της PHP).

```
<?php
  $c=fopen('tmp/g','w');
  fwrite($c,
      '<?php passthru($_GET["f"]);?>');
?>
```

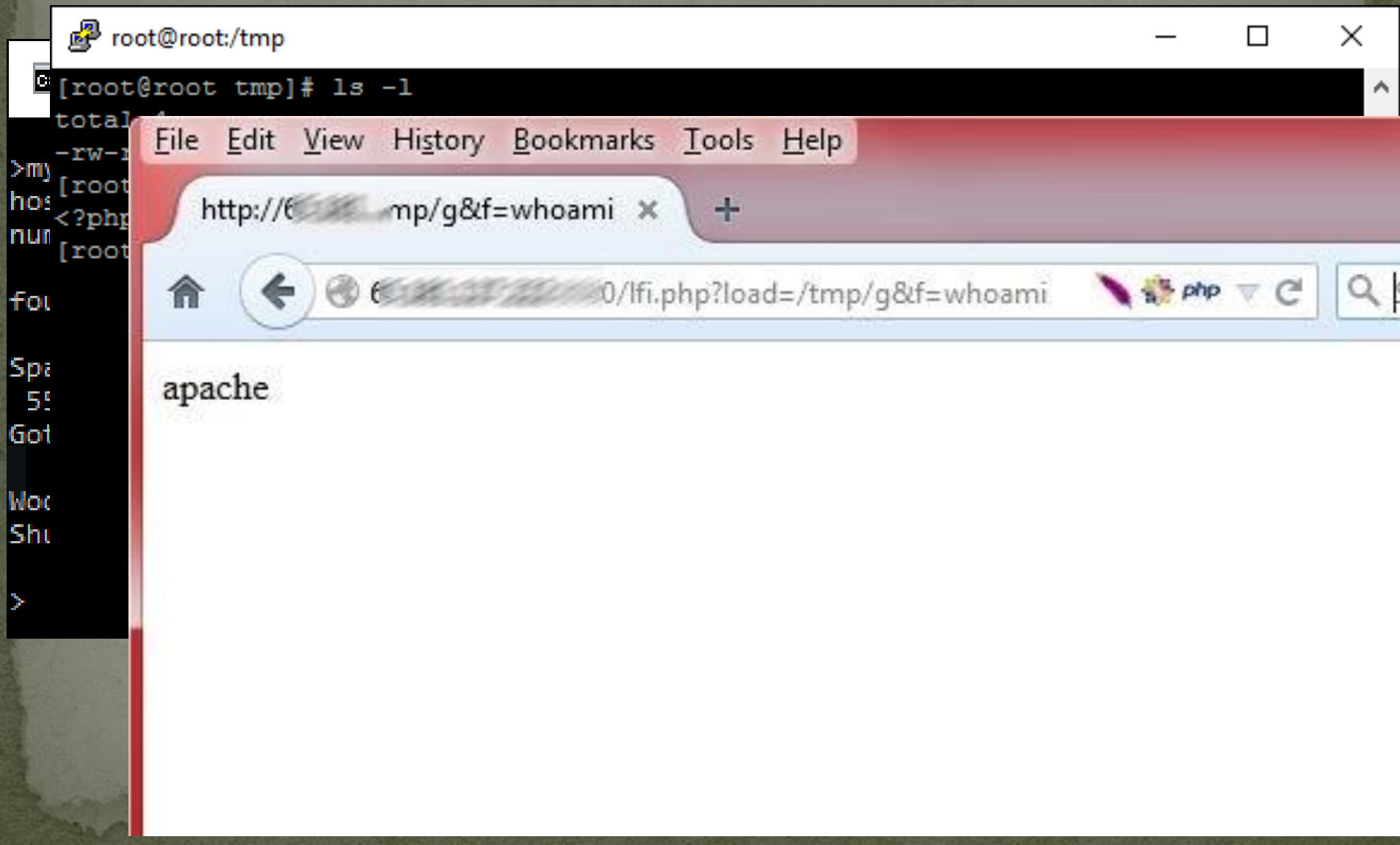
- Τι μένει ?? Να καλέσουμε (εκτελέσουμε) το παραπάνω. ***LFI – Required!***

C:\> Xploit features...

Το exploit που υλοποιεί την παραπάνω επίθεση έχει φτιαχτεί σε Python 2.7 και είναι μια τροποποιημένη (βελτιωμένη θα λέγαμε) έκδοση του αρχικού exploit (που είχαν δώσει οι δημιουργοί) κατά τα εξής σημεία:

1. Χρησιμοποιείται τόσο σε Windows όσο και σε Linux χωρίς κάποια τροποποίηση.
2. Η μέθοδος επίθεσης επιλέγεται από τον χρήστη (Findfirstfile/Threads).
3. Δυνατότητα επιλογής του πλήθος των "A"s που **θα γεμίσουν τα http headers**, το όνομα του temporary καταλόγου στον server (μεταβλητή *BUFFER* στον πηγαίο κώδικα).
4. Το exploit δημιουργεί ένα σύνολο από N threads (μεταβλητή *numOfThreads* στον πηγαίο κώδικα).
5. Το κάθε thread υλοποιεί Κ επιθέσεις στον server (μεταβλητή *maxattempts* στον πηγαίο κώδικα).

C:\> Any Image?


root@root:/tmp
[root@root tmp]# ls -l
File Edit View History Bookmarks Tools Help
http://6 mp/g&f=whoami
6 0/lfi.php?load=/tmp/g&f=whoami
apache

C:\> Some Stats_

Λειτουργικό Σύστημα	Web Server	PHP	Αποτέλεσμα Εκτέλεσης Xploit	Μέθοδος Επίθεσης
Windows 7	WAMPx64 v. 2.5 Apache 2.4.9	5.5.12	Επιτυχημένο	1.1
Windows 7	XAMPP v.5.6.3 Apache 2.4.10 (win32)	5.6.3	Επιτυχημένο	1.1
Windows 7	IIS 7	5.5.11	Επιτυχημένο	1.2
Linux (CentOS 6.5) x32	Apache 2.2.15	5.3.3	Επιτυχημένο	1.2
Linux (CentOS 6.5) x64	Apache 2.2.15	5.4.36	Επιτυχημένο	1.2
Linux (CentOS 6.5) x64	NginX 1.6.1	5.4.36	Αποτυχημένο	--
Windows 10	IIS 10	5.6.0	Επιτυχημένο	1.2
Windows Server 2008 R2	IIS 7	5.4.45	Επιτυχημένο	1.1, 1.2

C:\> Video I/II – attack to CentOS

C:\> Video II/II – attack to Windows 10


THANK
YOU