SECURING DISTRIBUTED SOFTWARE

DREW MCLELLAN, PHPSW SEPTEMBER 2016

DEALING WITH BUGS ON A PROJECT WITH RESIDENT DEVS IS EASY

FOR AN UPDATE TO A DEPENDANCY

- READ ABOUT AN UPDATE
- DOWNLOAD AND TEST THE PATCH
- PUSH IT TO PRODUCTION

FOR YOUR OWN CODE

- IDENTIFY THE PROBLEM
  - FIX THE PROBLEM
    - TEST THE PATCH
- PUSH IT TO PRODUCTION

DEALING WITH BUGS AS A LIBRARY MAINTAINER IS EASY

BUGS IN LIBRARIES

- IDENTIFY THE PROBLEM
- FIX THE PROBLEM
- TEST THE PATCH
- PUBLISH A NEW RELEASE
- ANNOUNCE THAT USERS SHOULD UPDATE

BUT WHAT HAPPENS FOR THINGS THAT FALL IN BETWEEN?

PERCH IS A SELF-HOSTED CONTENT MANAGEMENT SYSTEM

PERCH

- PHP + MYSQL
- (GD, IMAGICK ETC)
- THE WORDPRESS STACK

AN ANTIDOTE TO WORDPRESS

PERCH

- NO RELIANCE ON DOZENS OF THIRD PARTY PLUGINS
- STABLE AND RELIABLE. SET AND FORGET
- UPDATE WHEN YOU WORK ON THE SITE. NOT IN BETWEEN
- EXCELLENT SECURITY TRACK RECORD

PROJECT-WORK FRIENDLY

EXCELLENT SECURITY TRACK RECORD

FROM MAY 2009 TO NOVEMBER 2015: NO EXPLOITABLE SECURITY ISSUES

WE GOT AN EMAIL...

FROM A USER SAYING THEY COULD HACK OUR ONLINE DEMO SERVER AND DOWNLOAD A LIST OF ALL OUR CUSTOMERS AND WOULD WE PAY THEM FOR THIS INFORMATION?

WE KNEW TWO THINGS

1. THEY COULDN'T ACCESS OUR CUSTOMER DATA. AT MOST THEY COULD GET THE NAMES AND EMAIL ADDRESSES OF 100 OR SO CURRENT DEMO USERS (IT'S ALL SANDBOXED)
2. WE DON'T NEGOTIATE WITH TERRORISTS

THEY FAILED TO REALISE WE COULD JUST LOOK AT OUR LOGS

FROM THE LOGS

- I COULD SEE THEY'D UPLOADED A PHP WEBSHELL
- I COULD SEE HOW THEY'D UPLOADED IT. WHICH LEAD ME DIRECTLY TO THE BUG
- I COULD ALSO SEE THAT EVEN THOUGH THEY'D UPLOADED IT. OUR SERVER WOULDN'T RUN IT

WEBSHELL ATTACKS

WEBSHELL

I WASN'T FAMILIAR WITH THESE

SINGLE PAGE PHP SCRIPT THAT PROVIDES A WEB CONSOLE FOR EXECUTING SHELL COMMANDS AND CAUSING HAVOC

TWO THINGS NEED TO HAPPEN FOR THE ATTACK TO WORK

1. THE ATTACKER NEEDS TO BE ABLE TO GET THE SCRIPT ONTO THE SERVER
2. THE SCRIPT NEEDS TO BE ABLE TO EXECUTE VIA A STANDARD HTTP REQUEST

IT WAS POSSIBLE TO GET THE FILE ONTO THE SERVER

- OUR DEMO SERVER IS WELL CONFIGURED, SO IT FAILED AT THE SECOND PART
  - THE SCRIPT COULDN'T RUN

PERCH HAS PLUGINS FOR USING RICH EDITORS ON TEXTAREAS

F'KING WYSIWYG

- THESE CAN BE PLAIN TEXT, MARKDOWN, TEXTILE, OR HTML
- AS LONG AS THE EDITOR ENHANCES AN HTML TEXTAREA INPUT, IT'LL WORK

YOU NEED TO WRITE TWO THINGS

1. A BIT OF JAVASCRIPT TO INITIALISE THE EDITOR
2. IF THE EDITOR HAS A FILE UPLOAD BUTTON, A QUICK PHP SCRIPT TO HANDLE THE FILE UPLOAD VIA OUR API

UH OH. FILE UPLOADS.

FILE UPLOADS

THE DEFAULT EDITOR WE INCLUDE DOES MARKDOWN AND IS CALLED MARKITUP.

IT HAS A VERY QUICK. SIMPLE IMAGE AND FILE UPLOAD SCRIPT WHICH GRABS THE FILE USES THE PERCH API TO ADD IT AS CONTENT

THE PERCH API HAS TWO POSSIBLE ENTRY POINTS

ONE DESIGNED FOR ON-PAGE USE AS WEB PAGES ARE RENDERED

- LIGHT. FAST. DOES AS LITTLE WORK AS POSSIBLE
  - 'RUNTIME'

ONE DESIGNED FOR CONTROL PANEL OPERATIONS

- MORE COMPLEX, RICHER, DOES FAR MORE TO ASSIST BUT USES MORE RESOURCES
  - 'ADMIN'
  - CHECKS AUTHENTICATION
  - CHECKS AUTHENTICATION
  - CHECKS AUTHENTICATION

CHECKS AUTHENTICATION

THE MARKITUP UPLOAD SCRIPT WAS USING THE RUNTIME ENTRY POINT

- SHOULD HAVE BEEN USING THE ADMIN ENTRY POINT
- FILES COULD BE UPLOADED WITHOUT FIRST CHECKING THAT THE USER WAS AUTHENTICATED

ANYTHING COULD POST A FILE TO THE SCRIPT

AND IT WOULD UPLOAD IT...

... AND HELPFULLY GET A RESPONSE WITH THE URL TO THE UPLOADED FILE. JEEZ.

BUT WE'RE NOT
COMPLETELY DAFT

ALL UPLOADS ARE PROCESSED THROUGH A CENTRAL CLEARING HOUSE THAT DOES BASICS LIKE CHECKING MIME TYPES AND FILE EXTENSIONS

YOU CAN'T UPLOAD A FILE TO PERCH IF IT HAS A FILE EXTENSION WITH 'PHP' IN IT (`.php`. `.php3`. `.php5`).

THEY GET SAVED AS `.php.txt`

THIS HELPS TO ADDRESS THE SECOND PART OF THE ISSUE

THEY CAN UPLOAD THE FILE. BUT IT WON'T EXECUTE BECAUSE WE'VE DEFUSED THE FILE EXTENSION

LOOKING AT MY LOG FILES I COULD SEE THIS IN ACTION

```
webshell.php > webshell.php.txt
webshell.php5 > webshell.php5.txt
```

THE DAMAGE WAS:

1. ANYONE CAN UPLOAD FILES TO YOUR WEBSITE (BAD!)
2. BUT THEY COULD UPLOAD AND EXECUTE SCRIPTS (PHEW!)

LOOKING BACK THROUGH SOURCE CONTROL.

THIS PROBLEM HAD EXISTED FOR 6 YEARS

- APART FROM THIS ONE GUY, NO ONE SEEMED TO HAVE FOUND IT
  - I'D MISSED IT IN DOZENS OF CODE REVIEWS

SO...

- WE FIGURED THE BEST COURSE OF ACTION WAS TO FIX IT SWIFTLY AND MOVE ON WITHOUT DRAWING SPECIFIC ATTENTION TO IT.
- ENTRY IN THE CHANGE LOG SAID:

```
Fixes bugs in MarkItUp editor
```

TIME PASSES

A FEW WEEKS LATER WE GOT AN EMAIL FROM SOMEONE CLAIMING THEIR PERCH SITE HAD BEEN HACKED

WE GET THESE FROM TIME TO TIME - IT'S ALWAYS SOMETHING ELSE ON THE SERVER, OR THE SERVER ITSELF

EXCEPT THEN WE GOT ANOTHER WITH THE SAME SYMPTOMS

AND ANOTHER. BUT WITH MORE INFORMATION. A WEBSHELL HAD BEEN UPLOADED TO THE PERCH RESOURCES FOLDER WITH A `.phtml` EXTENSION.

.phtml

THIS WAS A NEW ONE ON ME
I HAD NO IDEA IT WAS USED FOR PHP

.phtml WAS THE FILE EXTENSION FOR

PHP 2

`.phtml` FILES COULD BE UPLOADED WITHOUT BEING DEFUSED

WHICH WAS NO BIG DEAL BECAUSE WE DON'T HAVE OUR WEB SERVERS CONFIGURED TO RUN RANDOM FILE EXTENSIONS WE'VE NEVER HEARD OF. SO NO PROBLEM.

... BUT WAIT.

SOME OF OUR CUSTOMERS USE REALLY TERRIBLE HOSTING

WHICH IS BADLY CONFIGURED
AND MIGHT BE EXPECTING PHP 2 FILES
TO BE UPLOADED

CRAP

THIS PUT US IN AN ODD SITUATION

- A BUG IN THE SOFTWARE WAS BEING EXPLOITED
  - BUT WE'D ALREADY FIXED IT WEEKS AGO
- BUT CUSTOMERS WEREN'T UPDATING THEIR SOFTWARE SO THEY WERE STILL VULNERABLE

THE VENN DIAGRAM OF CUSTOMERS ON BAD HOSTING AND CUSTOMERS UNLIKELY TO APPLY SOFTWARE UPDATES IS ALMOST A PERFECT CIRCLE

EVEN IF WE MADE A LOT OF NOISE ABOUT THIS, WE POSSIBLY WOULDN'T EVEN REACH THE PEOPLE AFFECTED, AND THEY MIGHT NOT BE IN A POSITION TO DO ANYTHING

WE HAD CUSTOMERS ON VERY OLD VERSIONS OF THE SOFTWARE

SOMETIMES UP TO 4 OR 5 YEARS OLD.

UPDATING TO THE CURRENT VERSION WOULD BE A CHARGEABLE PROJECT THAT WOULD REQUIRE PLANNING AND SCHEDULING.

SO WHAT DO YOU DO?

YOU MAKE 79 INDIVIDUAL PATCH FILES

WE PRODUCED A DROP-IN REPLACEMENT FILE FOR 79 PREVIOUS RELEASES OF PERCH

UPDATING A SITE WAS A CASE OF FINDING YOUR VERSION, DOWNLOADING THE PATCH, UPLOADING IT TO YOUR SITE

GUARANTEED TO CHANGE NOTHING ELSE BUT PATCHING THE BUG

THIS WAS IMPORTANT AS IT NEEDED TO BE SAFE TO DO TO A LIVE SITE

IF ANYTHING NEEDED TESTING, IT WOULD NEED TIME SCHEDULING AND IT WOULDN'T GET DONE

WE EMAILED CUSTOMERS AND STRESSED THAT THERE WAS AN IMPORTANT SECURITY UPDATE THAT NEEDED APPLYING STRAIGHT AWAY

DOWNSIDES

- GENERATING 79 PATCHES IS LABORIOUS
- WE HAVE NO WAY TO EASILY SEE IF A SITE IS PATCHED
- THAT ONE FILE DOESN'T AFFECT THE PART OF THE APP THAT REPORTS ITS VERSION NUMBER
- (WE COULD TACKLE THIS VARIOUS WAYS IF IT BECAME AND ISSUE)

UPSIDES

- A EASY FIX GAVE CUSTOMERS CONFIDENCE THAT WE WERE ON TOP OF THE PROBLEM
- A QUICK AND SAFE-TO-APPLY UPDATE MEANT THAT CUSTOMERS ACTUALLY UPDATED
- ...OR ENOUGH UPDATED THAT IT WASN'T WORTHWHILE ATTACKERS TO CONTINUE TO WASTE ENERGY ON

WHAT WE LEARNED

1. TAKE MINOR SECURITY BUGS SERIOUSLY

AS THEY CAN OPEN THE DOOR TO MAJOR ONES

2. MAKE IT REALLY EASY

IF YOU NEED PEOPLE TO UPDATE, ELSE THEY'LL PUT IT OFF

3. FINDING SERIOUS BUGS IN YOUR OWN CODE MAKES YOU FEEL REALLY BAD

BUT THAT'S OK. YOU SHOULD FEEL BAD.

THANKS!

JOIND.IN/TALK/8062B

@DREWM