EC-Council Certified Ethical Hacker v6.1

Cheat Sheet Exercises
How to Use the Cheat Sheets

Students often report that the most difficult thing about the CEH exam is the terms, tools, numbers, log files, packet dumps and example scripts. None of these items can be understood without the concepts that give them meaning, but once the concepts are clear, it is still necessary to be exposed to the raw data until they are second nature.

Cheatsheets are exercises that can be used to assist with memorization and refresh before the time of the exam. They are not comprehensive reference guides. They are designed to provide only enough data to trigger the memory or assess what needs to be better understood.

Having a list of everything at your fingertips is helpful on the job but is almost useless as a study tool. You must interact with the data in order to convert it to information and own it.

Since the exam is not open book, the goal is in fact to get to a point where you no longer need the cheat sheets at all.

Each cheat sheet is a concept object. These are examples to get you started and provide enough information to establish a grasp of the object at hand. Print them out, and hand copy each one in your own writing to another sheet of paper. Arrange the material in your own way, and add notes to them as you study.

Practice this at least three times. On the third try you may find you can copy the entire thing without looking at the original. Then you have mastered it, and will have problems recalling important data during the real exam.

In summary, to get the most out of these study aids, follow these simple tips:

1. Check back often for new versions
2. Print them out and copy them by hand to a blank piece of paper; three times.
3. Take additional notes, fill in any information that seems to be missing
# Chapter Map for the Cheat Sheets

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Topic</th>
<th>Subtopics</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Ethical Hacking</td>
<td>CEH Prerequisites</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Terms and Definitions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Methodologies</td>
</tr>
<tr>
<td>02</td>
<td>Hacking Laws</td>
<td>Legal Issues</td>
</tr>
<tr>
<td>03</td>
<td>Footprinting</td>
<td>Domain Name Service</td>
</tr>
<tr>
<td>04</td>
<td>Google Hacking</td>
<td>Google Hacking</td>
</tr>
<tr>
<td>05</td>
<td>Scanning</td>
<td>NMap Scan Types</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TCP Handshake</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ports and Protocols</td>
</tr>
<tr>
<td>06</td>
<td>Enumeration</td>
<td>Enumeration</td>
</tr>
<tr>
<td>07</td>
<td>System Hacking</td>
<td>Password Cracking</td>
</tr>
<tr>
<td>08</td>
<td>Trojans and Backdoors</td>
<td>Trojans and Malware</td>
</tr>
<tr>
<td>09</td>
<td>Virus and Worms</td>
<td>Virus Trivia</td>
</tr>
<tr>
<td>10</td>
<td>Sniffing, Spoofing, Hijacking</td>
<td>Sniffing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MAC Addresses</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Internet Protocol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Internet Control Message Protocol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>11</td>
<td>Social Engineering</td>
<td>Social Engineering</td>
</tr>
<tr>
<td>12</td>
<td>Denial of Service</td>
<td>DoS and DDoS Tools</td>
</tr>
<tr>
<td>13</td>
<td>Buffer Overflows</td>
<td>Buffer Overflows</td>
</tr>
<tr>
<td>14</td>
<td>Web Servers and Applications</td>
<td>HTTP and URLs</td>
</tr>
<tr>
<td>15</td>
<td>Wireless Networks</td>
<td>Wireless Technology</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wardriving</td>
</tr>
<tr>
<td>16</td>
<td>Cryptography</td>
<td>Cryptography</td>
</tr>
<tr>
<td>17</td>
<td>Hacking Linux</td>
<td>Linux Operating System</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Linux Commands</td>
</tr>
<tr>
<td>18</td>
<td>IDS, Firewalls, Honeypots</td>
<td>Firewalls and IPTables</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IDS and Snort</td>
</tr>
<tr>
<td>**</td>
<td>Misc Cheat Sheets</td>
<td>Command Line Tools</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Syntax Recognition</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Random Recall Exercise</td>
</tr>
</tbody>
</table>
CEH Prerequisites

There are entry level security classes, but security is not an entry level subject. In order to be comfortable with the CEH training, pre-requisites are assumed and test items will involve topics that time might not permit covering during the live training. Prior to training, try to refresh your skill in the following areas. The more time spent on this step the more comfortable the training experience will be.

**Know the basics of Information security**

- Concepts such as "CIA (Confidentiality, Integrity, Availability)
- Coverage would have come during CompTIA or CISSP training

**Know the basics of networking**

- Physical layer, cabling, hardware devices
- The function of switches, routers, firewalls
- IP Addressing, Subnetting and CIDR notation

**Know how to convert numbers**

- Decimal, Octal, Binary; in all directions and combinations

**Know the basics of Cryptography**

- There is a module in the class on Crypto, but there may not be time to cover it in class.
- Sufficient coverage would have come during CompTIA Security+ or CISSP

**Know the OSI model**

<table>
<thead>
<tr>
<th>Layer</th>
<th>Number</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>7</td>
<td>Service protocols</td>
</tr>
<tr>
<td>Presentation</td>
<td>6</td>
<td>Data formats</td>
</tr>
<tr>
<td>Session</td>
<td>5</td>
<td>Authentication, Cryptographic agreements</td>
</tr>
<tr>
<td>Transport</td>
<td>4</td>
<td>Ports, logical service to service connections</td>
</tr>
<tr>
<td>Network</td>
<td>3</td>
<td>Network to network delivery</td>
</tr>
<tr>
<td>Data Link</td>
<td>2</td>
<td>Host to host links, contention</td>
</tr>
<tr>
<td>Physical</td>
<td>1</td>
<td>Media</td>
</tr>
</tbody>
</table>

**Know how to use a Windows PC**

- Be familiar with the Windows Graphical User Interface
- Find toolbar icons, manage folders and files, use network shares
- The labs in this class are difficult and must move rapidly, slowdowns for poor PC skills may result in just watching the demonstration at times, please be understanding of this and courteous to the other students.
Read the following terms and make sure you know their meaning. Look up any that you are not comfortable with. On your own cheat sheet, jot down any additional terms you run across that struck you as new or odd.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hax0r</td>
<td>Hacker</td>
</tr>
<tr>
<td>Uberhacker</td>
<td>Good hacker</td>
</tr>
<tr>
<td>L33t Sp33k</td>
<td>Replacing characters to avoid filters</td>
</tr>
<tr>
<td>Full disclosure</td>
<td>Revealing vulnerabilities</td>
</tr>
<tr>
<td>Hacktivism</td>
<td>Hacking for a cause</td>
</tr>
<tr>
<td>Suicide Hacker</td>
<td>Hopes to be caught</td>
</tr>
<tr>
<td>Ethical Hacker</td>
<td>Hacks for defensive purposes</td>
</tr>
<tr>
<td>Penetration Test</td>
<td>Determine true security risks</td>
</tr>
<tr>
<td>Vulnerability Assessment</td>
<td>Basic idea of security levels</td>
</tr>
<tr>
<td>Vulnerability Researcher</td>
<td>Tracks down vulnerabilities</td>
</tr>
<tr>
<td>White hat</td>
<td>Hacks with permission</td>
</tr>
<tr>
<td>Grey hat</td>
<td>Believes in full disclosure</td>
</tr>
<tr>
<td>Black hat</td>
<td>Hacks without permission</td>
</tr>
<tr>
<td>White Box</td>
<td>A test everyone knows about</td>
</tr>
<tr>
<td>Grey Box</td>
<td>A test with a very specific goal but unspecific means</td>
</tr>
<tr>
<td>Black Box</td>
<td>A test no one knows is happening</td>
</tr>
<tr>
<td>Threat</td>
<td>Potential event</td>
</tr>
<tr>
<td>Vulnerability</td>
<td>Weakness</td>
</tr>
<tr>
<td>Exposure</td>
<td>Accessibility</td>
</tr>
<tr>
<td>Exploit</td>
<td>Act of attacking</td>
</tr>
<tr>
<td>TOE</td>
<td>Target of Evaluation</td>
</tr>
<tr>
<td>Rootkit</td>
<td>Hides processes that create backdoors</td>
</tr>
<tr>
<td>Botnet</td>
<td>Robot network that can be commanded remotely</td>
</tr>
<tr>
<td>Buffer Overflow</td>
<td>Hijack the execution steps of a program</td>
</tr>
<tr>
<td>Shrinkwrap Code</td>
<td>Reused code with vulnerabilities</td>
</tr>
</tbody>
</table>
**Methodologies**

This class tells a story, and understanding that story is far more important than memoriing these lists. Think about what actions are taken during each phase, and notice how they logically progress.

**The phases of an attack**

1. **Reconnaissance** Information gathering, physical and social engineering, locate network range
2. **Scanning - Enumerating** Live hosts, access points, accounts and policies, vulnerability assessment
3. **Gaining Access** Breech systems, plant malicious code, backdoors
4. **Maintaining Access** Rootkits, unpatched systems
5. **Clearing Tracks** IDS evasion, log manipulation, decoy traffic

**Information Gathering**

1. Unearth initial information What/ Who is the target?
2. Locate the network range What is the attack surface?
3. Ascertain active machines What hosts are alive?
4. Open ports / access points How can they be accessed?
5. Detect operating systems What platform are they?
6. Uncover services on ports What software can be attacked?
7. Map the network Tie it all together, document, and form a strategy.
**Legal Issues**

Be able to describe the importance of each of these items. The exam will not go into depth on this, just be prepared to identify the issues.

### United States

<table>
<thead>
<tr>
<th>Act / Legislation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer fraud and abuse act</td>
<td>Addresses hacking activities</td>
</tr>
<tr>
<td>18 U.S.C. 1029 Possession of Access Devices</td>
<td></td>
</tr>
<tr>
<td>18 U.S.C. 1030 Fraud and Related Activity in Connection with Computers</td>
<td></td>
</tr>
<tr>
<td>CAN-SPAM</td>
<td>Defines legal eMail marketing</td>
</tr>
<tr>
<td>SPY-Act</td>
<td>Protects vendors monitoring for licence enforcement</td>
</tr>
<tr>
<td>DMCA - Digital Milenium Copyright Act</td>
<td>Protects intellectual property</td>
</tr>
<tr>
<td>SOX - Sarbanes Oxley</td>
<td>Controls for corporate financial processes</td>
</tr>
<tr>
<td>GLBA - Gramm-Leech Bliley Act</td>
<td>Controls use of personal financial data</td>
</tr>
<tr>
<td>HIPPA - Health Information Portability and Protection Act</td>
<td>Privacy for medical records</td>
</tr>
<tr>
<td>FERPA - Family Educational Rights and Privacy Act</td>
<td>Protection for education records</td>
</tr>
<tr>
<td>FISMA - Federal Information Security Management Act</td>
<td>Government networks must have security standards</td>
</tr>
</tbody>
</table>

### Europe

<table>
<thead>
<tr>
<th>Act / Legislation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer misuse act of 1990</td>
<td>Addresses hacking activities</td>
</tr>
<tr>
<td>Human Rights Act of 1990</td>
<td>Ensures privacy rights</td>
</tr>
</tbody>
</table>
**Domain Name Service**

DNS is critical in the footprinting of a target network. It can sometimes save the attacker a lot of time, or at least corroborate other information that has been gathered. DNS is also a target for several types of attack.

*Fields in the SOA record: (Time in seconds)*

1882919 7200 3600 14400 2400
Serial Refresh Retry Expiry TTL

*Requesting a zone transfer*

nslookup; ls -d example.dom
dig @ns1.example.dom AXFR
host -t AXFR example.dom ns1.example.dom

*Using Whois*

whois example.dom

*Regional Internet Registrars*

ARIN (North America)
APNIC (Asia Pacific Region)
LACNIC (Southern and Central America and Caribbean)
RIPE NCC (Europe, the Middle East and Central Asia)
AfriNIC (Africa)

*Attacks against DNS servers*

Zone transfers Information gathering shortcut
Zone poisoning Breach the primary server and alter the zone file to corrupt the domain
Cache poisoning Send false answers to cache servers until they store them
Reflection DoS Send bogus requests into a chain of servers that do recursive queries
Google Hacking

An attacker will use Google to enumerate a target without ever touching it. The advanced search syntax is easy to use but can be quirky at times. It takes practice and experimentation.

Using Advanced Search

operator:keyword additional search terms

Advanced Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>site</td>
<td>Confines keywords to search only within a domain</td>
</tr>
<tr>
<td>ext</td>
<td>File extension</td>
</tr>
<tr>
<td>loc</td>
<td>Maps location</td>
</tr>
<tr>
<td>intitle</td>
<td>Keywords in the title tag of the page</td>
</tr>
<tr>
<td>allintitle</td>
<td>Any of the keywords can be in the title</td>
</tr>
<tr>
<td>inurl</td>
<td>Keywords anywhere in the URL</td>
</tr>
<tr>
<td>allinurl</td>
<td>Any of the keywords can be in the URL</td>
</tr>
<tr>
<td>incache</td>
<td>Search Google cache only</td>
</tr>
</tbody>
</table>

Keyword combinations

password | passlist | username | user
login | logon
Administrator | Admin | Root
Prototype | Proto | Test | Example

Examples

site:intenseschool.com (ceh ecsa lpt)
intitle:index.of
allinurl:login logon
Nmap Scan Types

Nmap is the de-facto tool for footprinting networks. It is capable of finding live hosts, access points, fingerprinting operating systems, and verifying services. It also has important IDS evasion capabilities.

Discovery Scans

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-sP</td>
<td>Ping</td>
</tr>
<tr>
<td>-sL</td>
<td>List Scan</td>
</tr>
<tr>
<td>-sO</td>
<td>Protocol</td>
</tr>
<tr>
<td>-sV</td>
<td>Verify</td>
</tr>
<tr>
<td>-sL</td>
<td>List scan</td>
</tr>
</tbody>
</table>

Normal Scans

<table>
<thead>
<tr>
<th>Option</th>
<th>Desc</th>
<th>Flags</th>
<th>Windows</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Open</td>
<td>Closed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Open</td>
<td>Closed</td>
</tr>
<tr>
<td>-sT</td>
<td>Connect</td>
<td>S</td>
<td>SA</td>
<td>RA</td>
</tr>
<tr>
<td>-sS</td>
<td>Stealth</td>
<td>S</td>
<td>SA</td>
<td>RA</td>
</tr>
</tbody>
</table>

Inverse Scans

<table>
<thead>
<tr>
<th>Option</th>
<th>Desc</th>
<th>Flags</th>
<th>Windows</th>
<th>Linux</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Open</td>
<td>Closed</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Open</td>
<td>Closed</td>
</tr>
<tr>
<td>-sN</td>
<td>Null</td>
<td>-</td>
<td>RA</td>
<td>RA</td>
</tr>
<tr>
<td>-sX</td>
<td>Xmas</td>
<td>UPF</td>
<td>RA</td>
<td>RA</td>
</tr>
<tr>
<td>-sF</td>
<td>Fin</td>
<td>F</td>
<td>RA</td>
<td>RA</td>
</tr>
<tr>
<td>-sA</td>
<td>Ack</td>
<td>A</td>
<td>R</td>
<td>R</td>
</tr>
<tr>
<td>-sW</td>
<td>Window</td>
<td>A</td>
<td>R</td>
<td>R</td>
</tr>
</tbody>
</table>

Other Important Nmap Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-A</td>
<td>Enable OS detection, Version detection, Script scanning and Traceroute</td>
</tr>
<tr>
<td>-n</td>
<td>Do not lookup DNS</td>
</tr>
<tr>
<td>-v</td>
<td>Verbose output</td>
</tr>
<tr>
<td>-T [0-5]</td>
<td>Timing - 5 is faster</td>
</tr>
<tr>
<td>-P0</td>
<td>Do not ping first</td>
</tr>
</tbody>
</table>
**TCP Flags**

This test will have scenarios that require you demonstrate an understanding of TCP behavior including Nmap scan types. Be sure to know each of these combinations well.

**TCP Flags**

0 0 URG ACK PSH RST SYN FIN

**TCP Handshake (Open Port)**

<table>
<thead>
<tr>
<th>Direction</th>
<th>Binary</th>
<th>Hex</th>
<th>Flags</th>
<th>Seq</th>
<th>Ack</th>
</tr>
</thead>
<tbody>
<tr>
<td>A -&gt; B</td>
<td>00000010</td>
<td>0x02</td>
<td>S</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B -&gt; A</td>
<td>00010010</td>
<td>0x12</td>
<td>A S</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>A -&gt; B</td>
<td>00010000</td>
<td>0x10</td>
<td>A</td>
<td>2</td>
<td>11</td>
</tr>
</tbody>
</table>

**TCP Handshake (Closed Port)**

<table>
<thead>
<tr>
<th>Direction</th>
<th>Binary</th>
<th>Hex</th>
<th>Flags</th>
<th>Seq</th>
<th>Ack</th>
</tr>
</thead>
<tbody>
<tr>
<td>A -&gt; B</td>
<td>00000010</td>
<td>0x02</td>
<td>S</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>B -&gt; A</td>
<td>00010100</td>
<td>0x14</td>
<td>A R</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

**NMap Stealth Scan (Open Port)**

<table>
<thead>
<tr>
<th>Direction</th>
<th>Binary</th>
<th>Hex</th>
<th>Flags</th>
<th>Seq</th>
<th>Ack</th>
</tr>
</thead>
<tbody>
<tr>
<td>A -&gt; B</td>
<td>00000010</td>
<td>0x02</td>
<td>S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B -&gt; A</td>
<td>00010010</td>
<td>0x12</td>
<td>A S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A -&gt; B</td>
<td>00000100</td>
<td>0x04</td>
<td>R</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NMap Xmas Scan (Open Port)**

<table>
<thead>
<tr>
<th>Direction</th>
<th>Binary</th>
<th>Hex</th>
<th>Flags</th>
<th>Seq</th>
<th>Ack</th>
</tr>
</thead>
<tbody>
<tr>
<td>A -&gt; B</td>
<td>00101001</td>
<td>0x29</td>
<td>U P F</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

No response from Linux hosts, R A from Windows

**NMap ACK Scan**

<table>
<thead>
<tr>
<th>Direction</th>
<th>Binary</th>
<th>Hex</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>A -&gt; B</td>
<td>00010000</td>
<td>0x10</td>
<td>A</td>
</tr>
<tr>
<td>A -&gt; B</td>
<td>00000100</td>
<td>0x04</td>
<td>R</td>
</tr>
</tbody>
</table>

Solaris will not respond on open ports
Ports and Protocols

These must be memorized! Also be prepared to convert them to hexadecimal representation in case they must be identified in a packet dump, log file, IDS rule, or a sniffer capture/display filter.

Protocols

1  ICMP
6  TCP
17  UDP
47  GRE
50  AH
51  ESP

Ports

20 - 21  FTP
22  SSH
23  Telnet
25  SMTP
42  WINS
53  DNS
80 - 81 -8080  HTTP
88  Kerberos
110  POP3
111  Portmapper (Linux)
119  NNTP
135  RPC-DCOM
137 - 138 - 139  SMB
143  IMAP
161 - 162  SNMP
389  LDAP
445  CIFS
1080  SOCKS5
3389  RDP
6667  IRC
14237  Palm Pilot Remote Sync

Trojan Horses

7777  Tini
12345  NetBus
27374  Back Orifice
31337  Sub7
 Enumeration

Enumeration is the act of making a list of policies, user accounts, shares and other resources. This step happens just before vulnerability assessment and helps the attack put together the best strategy for gaining access.

 Establishing a Null Session

net use \\[target ip]\IPC$ "" /user:"

 Protecting Information Disclosure

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\LSA\RestrictAnonymous

"0" is the default for Windows 2000 and gives up everything
"1" is the default for Windows 2003 and gives up less
"2" is the most secure setting but makes a machine not very cooperative with others

Microsoft SIDs

S-1-5-21-<……….>-500 Built-in Local administrator
S-1-5-21-<……….>-501 Built-in Local guest
S-1-5-21-<……….>-512 Built-in Domain administrator
S-1-5-21-<……….>-1000 Anything above 1000 are users that have been created

Ports involved with enumerations attacks

111 Linux Portmapper Service
42 WINS
88 Kerberos
135 Windows RPC-DCOM
137 NetBIOS Name Service
138 NetBIOS Datagram Service
139 NetBIOS Sessions
161 SNMP Agent
162 SNMP Traps
389 LDAP
445 CIFS (Common Internet File System)

Misc.

"public" and "private" default community SNMP strings
1.1.1.2.1.0.0.1.3.4.1.4 is an SNMP OID
ou=sales,cn=example... is an LDAP (LDIF) name string
fingerd the finger daemon was used in older UNIX systems
Password Cracking

This test will have scenarios that require you demonstrate an understanding of TCP behavior. Be sure to know each of these combinations well.

**Types of password cracking techniques**

- **Guessing**: Is the most efficient, assuming information gathering beforehand.
- **Dictionary**: Based on a predetermined list of words.
- **Brute Force**: Trying every possible combination of characters.
- **Hybrid**: A combination of all other attacks.

**LM Hashes**

Every password is ultimately 14 characters long, split into two 7 character halves. Passwords that are less than 7 characters are easily identified in the SAM file (hash ends in 404EE).

**Rainbow Tables**

"Time / Memory Trade off" Less memory than a lookup, less computing than a brute force. The hash is a way to combat rainbow tables.

**Cracking Effort**

- Weak passwords: can be cracked in seconds
- Strong passwords: might take the lifetime of several universes to crack
- Rainbow Tables: Solve the "Time / Memory Trade Off"
- DNA: Distributed Network Architecture

**Popular Cracking Tools**

- John the Ripper: Command line tool that runs under both Windows and Linux
- L0phtcrack: Commercial tool
- 0phtrcack: Open source tool that supports rainbow tables
- Cain and Abel: Powerful multipurpose tool that than sniff and crack passwords of many types
Trojans and Malware

The official definition is: A legitimate application that has been modified with malicious code. A Trojan horse is a social engineering technique. It masquerades as a legitimate download and injects the victim's host with an access point, or a client that can connect outbound to a server waiting remotely. They don't necessarily exploit a vulnerability unless privilege escalation is necessary. They provide a command environment for whoever connects to them that includes: File browsers, keyloggers, web cam viewer, and many additional tools.

Terms

Wrapper or Binder | Application used to combine a malicious binary and a legitimate program
Rootkit | Can be installed via Trojan, used to hide processes that create backdoor access
HTTP Trojan | Reverses a connection outbound through an HTTP or SHTTP tunnel
Netcat | Not really a Trojan, but often used in Trojan code to setup the listening socket
Hoax | Many legit tools are rumored to be Trojans but might not be
Keylogger | Records the keystrokes on the install host and saves them in a log

Famous Trojans

Tini | Small 3Kb file, uses port 7777
Loki | Used ICMP as a tunneling protocol
Netbus | One of the first RATs (Remote Authentication Trojan)
Sub 7 | Written in Delphi, expanded on what Netbus had demonstrated
Back Orifice | First modular malware, had the capabilities to be expanded on by outside authors
Beast | All in one Client / Server binary
MoSucker | Client could select the infection method for each binary
Nuclear RAT | Reverse connecting Trojan
Monkey Shell | Provides a powerful shell environment that can reverse connections and encrypt commands.

Detecting Trojans

netstat / fport | Command line tools for viewing open ports and connections
tcpview | GUI tool for viewing open ports and connections
Process Viewer | GUI tool for showing open processes including child processes
Autoruns | Lists all programs that will run on start up and where they are called from
Hijack This | Displays a list of unusual registry entries and files on the drive
Spybot S&D | Originally volunteer supported scanning and detection tool
Virus Trivia

No one is expecting you the student to stay on top of the 40k or so known malware variants that have been discovered. But there are a few that are significant for demonstrating the capabilities of this method of attack. Think of the malware mentions in the course as examples of what thousands of others have copied or improved upon.

Phases of an outbreak

Infection -> Spreading -> Attack

Virus Lifecycle

Design - > Replication - > Launch - > Detection - > Incorporation - > Elimination

Types of Viruses

- Boot Virus: Infects the boot sector of floppies or hard disks
- Macro Virus: Written in Microsoft Office Macro language
- Network Virus: Spreads via network shares
- Stealth Virus: Hides in a file, copies itself out to deliver payload
- Polymorphic Virus: Encrypts itself
- Cavity Virus: Hides in the empty areas of executables
- Tunneling Virus: Trace interceptor programs that monitor OS Kernel requests
- Camouflage Virus: Disguise themselves as legit files
- Multipartite Virus: Infects via multiple vectors
- Metamorphic Virus: Rewrites itself

Famous Viruses

- Elk Cloner: 1st virus
- Morris: 1st worm
- I Love You: VBScript worm, sent via email
- Melissa: Macro virus
- Klez: Mass mailer with its own SMTP engine
- Slammer: Targets SQL server, total size of 376 bytes
- MyDoom: Mass mailer, uses port 3127, attacks the hosts file
- MonteCarlo: Memory resident, copies to the end on exe files
Sniffing

Social Engineering is the most powerful attack tool. It requires no equipment or technology, and often minimal expense. Only proper user education and awareness can prevent it and even then, errors in judgment can still be exploited.

Methods for defeating a switch

Admin the switch  If the password for the switch can be guessed, a port can be placed into monitor mode
MAC Spoofing  Set the MAC address of a NIC to the same value as another
MAC Flooding  Overwhelm the CAM table of the switch so it coverts to hub mode
ARP Poisoning  Inject incorrect information into the ARP caches of two or more endpoints.

Wireshark command line tools

tshark  Command line version of Wireshark
dumpcap  Captures traffic
capinfos  Reads a saved capture file and returns statistics about it
editcap  Edit and/or translate the format of capture files
mergecap  Merges multiple capture files into one
text2pcap  Generates a capture file from an ASCII hexdump of packets
tcpflow  Extracts data streams from dump files
tcptrace  Analyzes TCP conversations
tcpreplay  Can resend capture packets

TCPDump capture filters

Capture filters will be kept simple on the test. They look basically like English phrases. Analyze the examples below to get an idea.

host www.example.com and not (port 80 or port 25)
port not 53 and not arp
ip proto 1

Wireshark display filters

Display filters work basically like: proto.field operator value

Analyse the following examples:

tcp.flags == 0x29
ip.addr != 192.168.1.1
tcp.port eq 25 or icmp
ip.src==192.168.0.0/16 and ip.dst==192.168.0.0/16
http.request.uri matches "login.html"
MAC Addresses

Sniffing and defeating Ethernet switches requires an understanding of hardware addresses. Due to the risks involved with these local attacks, Intrusion Detection Systems are looking for too much ARP traffic or strange MAC addresses.

The MAC 48 Format

A Media Access Control address is 48 bits
The first 3 bytes of the MAC is a vendor code
The other three bytes are arbitrarily assigned

A broadcast MAC address is

FF:FF:FF:FF:FF:FF

Addresses can be assigned in two ways

BIA - Burned in Address
OUI - Organizationally Unique Identifier

The two least significant bits of the first byte in the OUI address

nnnnnn0n = Universally administered address
nnnnnn1n = Administratively assigned
nnnnnn0n = Unicast traffic
nnnnnn1n = Multicast traffic
Internet Protocol

Internet protocol is responsible for packaging datagrams for delivery between networks. It is a "best effort" protocol with no error control or correction. For more information read RFC 791

Internet Protocol Header

```
0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|Version| IHL |Type of Service| Total Length |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Identification |Flags| Fragment Offset |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Time to Live | Protocol | Header Checksum |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Source Address |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Destination Address |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Options | Padding |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Example Internet Datagram Header

Checklist of items to concentrate on:

- How IPIDs work
- How the fragmentation works
- How the TTL works
- Protocol IDs
- Basic IP addressing principles
- DoS attacks relating to IP
Internet Control Message Protocol

ICMP is a transport protocol that creates message datagrams that can be exchanged by network hosts for troubleshooting, error reporting, and information. For more information read RFC 792. For a complete list of type and codes visit http://www.spirit.com/Resources/icmp.html

ICMP Header Example:

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Type   |    Code    |          Checksum          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  unused                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Internet Header + 64 bits of Original Data Datagram |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Echo Reply</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>Destination Unreachable</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>Administratively Prohibited</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>Echo Request</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>Redirect</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>Time Exceeded</td>
</tr>
<tr>
<td>13</td>
<td>-</td>
<td>Timestamp Request</td>
</tr>
</tbody>
</table>

Don't forget!!

Type 3 Code 13 means administratively prohibited
**User Datagram Protocol**

User Datagram Protocol is a simple fast transport protocol that is used for its low overhead in situations where error correction and flow control is not needed, such as short bursts of messages. UDP is difficult to firewall off effectively because it is stateless. For more information read RFC 768

---

**User Datagram Protocol**

```
  0    7    8   15   16   23   24   31
+----------------------------------+
|     Source       |   Destination    |
|     Port        |     Port         |
+----------------------------------+
|             |                |
|     Length    |   Checksum       |
+----------------------------------+
|                                  |
|     data octets    ...           |
+-----------------------
```

User Datagram Header Format

---

**Checklist of items to concentrate on:**

- Port addresses and ranges
- How ICMP and UDP assist each other
- UDP based Denial of Service Attacks
Transmission Control Protocol

TCP provides guaranteed transport and flow control of layer 5-7 messages. Along with IP, ICMP, and UDP, a good solid understanding of this protocol is critical for understanding: Scanning, Firewalls, Intrusion Detection, and various types of DoS attacks. For more information read RFC 793

**Transmission Control Protocol**

```
0    1    2    3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|            Source Port            | Destination Port |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                     Acknowledgment Number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|            Data            |
|       U|A|F|R|S|F|       |
| Offset| Reserved |R|C|S|S|Y|I|      Window |
|       |             |G|K|E|T|N|N|       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|       Checksum       |
|                       Urgent Pointer |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| Options |
|       Padding       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|       data       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

TCP Header Format

**Checklist of items to concentrate on:**

- Port addresses and ranges
- Order of the six flags
- How the handshake works
- How the sequence numbers work
- How session hijacking works
- Denial of service attacks related to TCP
Social Engineering

Social Engineering is the most powerful attack tool. It requires no equipment or technology, and often minimal expense. Only proper user education and awareness can prevent it and even then, errors in judgment can still be exploited.

The principles of Social Engineering

- Authority: An intimidating presence
- Scarcity: Create the perception of loss or lack of access to a resource
- Liking: Charm and charisma
- Reciprocation: The victim believes they owe the attacker a favor
- Consistency: Appealing the victim's true feelings and opinions
- Social Validation: Compliments and praise

Types of Social Engineers

- Insider Associates: Have limited authorized access, and escalate privileges from there.
- Insider Affiliates: Are insiders by virtue of an affiliation, they spoof the identity of the insider.
- Outsider Affiliates: Are non-trusted outsiders that use an access point that was left open.
DoS and DDoS

Denial of Services and Distributed Denial of Service attacks are embarrassing and inconvenient. They are extremely difficult to prevent from being attempted. The best defense is a well designed network that is hard to overwhelm.

DoS Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer Overflows</td>
<td>Crashes applications or services</td>
</tr>
<tr>
<td>Smurf</td>
<td>Spoofed traffic sent to the broadcast address of a network</td>
</tr>
<tr>
<td>Fraggle</td>
<td>UDP version of the Smurf, usually bouncing Chargen traffic off Echo ports</td>
</tr>
<tr>
<td>Ping of Death</td>
<td>Packet larger than the 64k limit</td>
</tr>
<tr>
<td>Teardrop</td>
<td>Offset values modified to cause fragments to overlap during reassembly, results in short packet</td>
</tr>
<tr>
<td>Unnamed</td>
<td>Offset values modified to cause gaps between fragments, results in long packets</td>
</tr>
<tr>
<td>Syn Flood</td>
<td>SYN flags sent to open ports, no completion of the handshake</td>
</tr>
<tr>
<td>Land</td>
<td>Traffic sent to a victim spoofing itself as the source, results in ACK storms</td>
</tr>
<tr>
<td>Winnuke</td>
<td>Sends TCP traffic with the URG flag set, causes CPU utilization to peak</td>
</tr>
</tbody>
</table>

Dos Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jolt2</td>
<td>Floods with invalid traffic results in 100% CPU utilization</td>
</tr>
<tr>
<td>Land and La Tierra</td>
<td>Executes teardrop and land attacks</td>
</tr>
<tr>
<td>Targa</td>
<td>Provides a menu of several DoS attacks</td>
</tr>
<tr>
<td>Blast20</td>
<td>Also considered to be a web server load tester</td>
</tr>
<tr>
<td>Crazy Pinger</td>
<td>ICMP flooder</td>
</tr>
<tr>
<td>UDP Flood</td>
<td>UDP flooder written by Foundstone</td>
</tr>
</tbody>
</table>

DDos Attacks

Botnets - Command and Control Center communicates with "Handlers" which in turn communicate with Zombies. The handlers and zombies are machines infected with malware. The C&CC is either a chatroom on IRC, or can even be a distributed system of infected machines.

DDoS Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trinoo</td>
<td>One of the first to demonstrate &quot;Master/slave&quot; DDoS attacks</td>
</tr>
<tr>
<td>Tribal Flood Network</td>
<td>Could launch several DoS attacks from distributed positions at the same time</td>
</tr>
<tr>
<td>TFN2K</td>
<td>Bug fixes and updates to the original TFN</td>
</tr>
<tr>
<td>Stacheldraht</td>
<td>Means &quot;Barbed Wire&quot; in German</td>
</tr>
<tr>
<td>Agobot</td>
<td>A modular IRC bot, many derivatives have been created from this code</td>
</tr>
<tr>
<td>Nuclear Bot</td>
<td>Developed by &quot;Nuclear Winter Crew&quot; and written in Delphi, many features</td>
</tr>
</tbody>
</table>
Buffer Overflows

It isn't necessary to become a "C" programmer to pass the test, but several basic concepts and terms are critical in the understanding of BO scripts and the detection of BO attacks.

**Terminology**

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stack</td>
<td>Memory place for short term processing</td>
</tr>
<tr>
<td>Heap</td>
<td>Memory space for long term program execution</td>
</tr>
<tr>
<td>Push</td>
<td>&quot;Push&quot; new instructions onto the stack</td>
</tr>
<tr>
<td>Pop</td>
<td>&quot;Pop&quot; instructions off the stack when processed</td>
</tr>
<tr>
<td>EIP</td>
<td>Execute Instruction Pointer, memory address of next instruction to be executed</td>
</tr>
<tr>
<td>NOOP</td>
<td>A &quot;do nothing&quot; instruction that wastes a clock cycle</td>
</tr>
<tr>
<td>NOOP Sled</td>
<td>Placed in a buffer overflow exploit to aid in running the payload</td>
</tr>
</tbody>
</table>

**Dangerous Functions**

The following functions are dangerous because they do not check the size of the destination buffers:

- `gets()`
- `strcpy()`
- `strcat()`
- `printf()`
- The `>>` operator is also dangerous for the same reason

**Canary bytes**

String terminating characters:

<table>
<thead>
<tr>
<th>Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LF</td>
<td>Line Feed</td>
</tr>
<tr>
<td>CR</td>
<td>Carriage Return</td>
</tr>
<tr>
<td>NULL</td>
<td>Null</td>
</tr>
<tr>
<td>EOF</td>
<td>End of File</td>
</tr>
</tbody>
</table>

A randomly chosen value can also be placed at the end of a stack and checked.

**Recognizing a buffer overflow attempt**

Apr 5 02:02:09 [3432] : nops: 62.32.54.123:3211 -> 192.168.3.4:135
0x90/0x90/0x90/0x90/0x90/0x90/0x90/0x90/0x90/0x90/
HTTP and URLs

HTTP is the protocol for the World Wide Web. The client (web browser) sends request to the server (Apache, IIS) which is turn passes the request to an application. There are several attack types that are possible in this exchange since all of these components can have vulnerabilities.

HTTP Error Codes

<table>
<thead>
<tr>
<th>Series</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td></td>
<td>Everything is OK</td>
</tr>
<tr>
<td>400</td>
<td></td>
<td>Could not provide requested resource (page not found, moved, authentication failure)</td>
</tr>
<tr>
<td>500</td>
<td></td>
<td>Could not process request (script error, database connection error)</td>
</tr>
</tbody>
</table>

ASCII Characters

- . %2E
- / %2F
- < %3C
- > %3E

Uniform Resource Locators (URL)

Protocol       FQDN         Resource Path                 Query String
http://www.example.com/folder/directory/page.asp?var=something&foo=some+other+thing

Representing IP Addresses

<table>
<thead>
<tr>
<th>Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dotted Quad</td>
<td><a href="http://192.168.100.125">http://192.168.100.125</a></td>
</tr>
<tr>
<td>Hex Quad</td>
<td><a href="http://0xC0.0xA8.0x64.0x7D">http://0xC0.0xA8.0x64.0x7D</a></td>
</tr>
<tr>
<td>Decimal</td>
<td><a href="http://3232261245">http://3232261245</a></td>
</tr>
</tbody>
</table>

Converting Dotted Quad to Decimal (using above example)

192.168.100.125

Formula  
\[(256^3 * 192) + (256^2 * 168) + (256^1 * 100) + (256^0 * 125)\]
Simplified  
\[16777216 * 192 + 65536 * 168 + 256 * 100 + 125\]
Simplified again  
\[3221225472 + 11010048 + 25600 + 125\] =
Answer  
3232261245
**Wireless Technology**

Wireless is fast becoming the network technology of choice because it is cheap and easy. It is also a hubbed environment that can leak signals for miles. Configuring wireless technologies is an often misunderstood process, and often leaves many opportunities available for attack.

**802.11**

<table>
<thead>
<tr>
<th>Spec</th>
<th>Distance</th>
<th>Speed</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11a</td>
<td>30M</td>
<td>54Mbps</td>
<td>5Ghz</td>
</tr>
<tr>
<td>802.11b</td>
<td>100M</td>
<td>11Mbps</td>
<td>2.4Ghz</td>
</tr>
<tr>
<td>802.11g</td>
<td>100M</td>
<td>54Mbps</td>
<td>2.4Ghz</td>
</tr>
<tr>
<td>802.11n</td>
<td>125M</td>
<td>600Mbps</td>
<td>5Ghz</td>
</tr>
</tbody>
</table>

802.11i is a rewrite of WEP called WPA/TKIP

**Wireless Security**

- **WEP**
  - Uses RC4 for the stream cipher with a 24b initialization vector
  - Key sizes are 40b or 104b
- **WPA**
  - Uses RC4 for the stream cipher but supports longer keys
- **WPA/TKIP**
  - Changes the IV with each frame and includes key mixing
- **WPA2**
  - Uses AES as the stream cipher and includes all the features of TKIP
- **OSA**
  - Open Systems Authentication is a non-protected AP that broadcasts its SSID
- **PSK**
  - Pre-Shared Key is protected by an encryption standard

**Terms and Tools**

- **Wardriving**
  - Driving around with portable equipment and locating wireless networks
- **Warchalking**
  - Writing symbols on the sidewalk or buildings communicating found networks
- **Jamming**
  - Producing white noise signals that overpower the Wifi networks
- **Netstumbler**
  - Finds wireless networks, SSIDS, and channels
- **Ministumbler**
  - for the pocket pc
- **Macstumbler**
  - for the Macintosh
- **AirPcap**
  - Hardware tools for wardriving, WEP cracking, and sniffing
- **Airopeek**
  - Sniffer that specializes in wireless traffic
- **AircrackNG**
  - WEP cracker
- **Airsnort**
  - Another WEP cracker
- **CoWPAtty**
  - WPA offline brute force cracker
Wireless Technology

Wireless is fast becoming the network technology of choice because it is cheap and easy. It is also a hubbed environment that can leak signals for miles. Configuring wireless technologies is an often misunderstood process, and often leaves many opportunities available for attack.

WLAN Channels

Each channel increments by .005Mhz

![WLAN Channels Diagram]

Wardriving Symbols

- **open node**
  - ssid
  - bandwidth

- **closed node**
  - ssid

- **WEP node**
  - ssid
  - access
  - contact
  - bandwidth
Cryptography

Cryptography is assumed pre-requisite for this class. It's still a good idea to review some core terminology before the exam.

Terms and Definitions

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plaint Text</td>
<td>The data set before encryption</td>
</tr>
<tr>
<td>Cipher Text</td>
<td>The result of encryption</td>
</tr>
<tr>
<td>Cryptanalysis</td>
<td>Attempting to “break” and encryption algorithm</td>
</tr>
<tr>
<td>Cryptography</td>
<td>Obscuring the meaning of a message</td>
</tr>
<tr>
<td>Steganography</td>
<td>Hiding a message within another</td>
</tr>
<tr>
<td>Salt</td>
<td>Ensures different keys are created each time</td>
</tr>
<tr>
<td>Initialization Vector</td>
<td>Change the characteristics of the key each time it is reused</td>
</tr>
</tbody>
</table>

Types of Cryptography

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric</td>
<td>Single key both encrypts and decrypts</td>
</tr>
<tr>
<td>Asymmetric</td>
<td>A pair of keys, public and private are mathematically associated</td>
</tr>
<tr>
<td>One-Way Hash</td>
<td>Cannot be reversed, only brute forced</td>
</tr>
<tr>
<td></td>
<td>Used to represent data, sometimes called &quot;Digital Fingerprint&quot; or &quot;Message Digest&quot;.</td>
</tr>
</tbody>
</table>

Symmetric Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES</td>
<td>Block</td>
<td>56 bit key used in LM Hash password storage</td>
</tr>
<tr>
<td>3DES</td>
<td>Block</td>
<td>128 bit key used in NTLM</td>
</tr>
<tr>
<td>RC4</td>
<td>Stream</td>
<td>Used in WEP</td>
</tr>
<tr>
<td>AES</td>
<td>Stream</td>
<td>Used in WPA2</td>
</tr>
</tbody>
</table>

Asymmetric Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA</td>
<td>Asymmetric</td>
<td>Used in SSL/TLS</td>
</tr>
<tr>
<td>Elliptic Curve</td>
<td>Asymmetric</td>
<td>Used in TLS for portable devices</td>
</tr>
</tbody>
</table>

One-Way Hashes

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD5</td>
<td>One Way Hash</td>
<td>128b hash value, used for integrity checks</td>
</tr>
<tr>
<td>SHA-1</td>
<td>One Way Hash</td>
<td>160b hash value, stronger than MD5</td>
</tr>
</tbody>
</table>
**Linux Operating System**

While it is not necessary to be a Linux administrator or developer to pass this test, there is some assumed knowledge of a few basics, particularly pertaining to Security issues.

**Linux File System**

- `/`: Root of the file system
- `/var`: Variable data, log files are found here
- `/bin`: Binaries, commands for users
- `/sbin`: System Binaries, commands for administration
- `/root`: Home directory for the root user
- `/home`: Directory for all home folders for non-privileged users
- `/boot`: Stores the Linux Kernel image and other boot files
- `/proc`: Direct access to the Linux kernel
- `/dev`: Direct access to hardware storage devices
- `/mnt`: Place to mount devices on onto user mode file system

**Identifying Users and Processes**

- INIT process ID: 1
- Root UID, GID: 0
- Accounts for services: 1-999
- All other users: Above 1000

**MAC Times**

- Modify: Modify the contents of the file
- Access: When the files was accessed last
- Change: Metadata change

Use the "touch -mac filename" command to update all of them at the same time

**Permissions**

<table>
<thead>
<tr>
<th></th>
<th>User</th>
<th>Group</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>400</td>
<td>040</td>
<td>004</td>
</tr>
<tr>
<td>W</td>
<td>200</td>
<td>020</td>
<td>002</td>
</tr>
<tr>
<td>X</td>
<td>100</td>
<td>010</td>
<td>001</td>
</tr>
<tr>
<td>SUID</td>
<td>4000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SGID</td>
<td></td>
<td></td>
<td>2000</td>
</tr>
</tbody>
</table>

**Examples**

- User can RWX, Group can RW and Others can R: 764
- User can RW, Group can R and others can R: 644
- SUID bit set, User and group can RWX: 4770
- SUID and GUID bit set, all users can RWX: 6777
# Linux Commands

Practice the following commands and be able to recognize them in a shell script or log file. Always remember to "manpage" a command. Get used to reading about options and usage.

<table>
<thead>
<tr>
<th>Command</th>
<th>Notable Options</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>man</td>
<td>/</td>
<td>Manual pages</td>
</tr>
<tr>
<td>ls</td>
<td>-l</td>
<td>Looksee into a directory</td>
</tr>
<tr>
<td>cd</td>
<td></td>
<td>Change directory</td>
</tr>
<tr>
<td>pwd</td>
<td></td>
<td>Print working directory</td>
</tr>
<tr>
<td>touch</td>
<td>-macr</td>
<td>Create a file or update its attributes</td>
</tr>
<tr>
<td>mv</td>
<td></td>
<td>Move a file</td>
</tr>
<tr>
<td>rm</td>
<td></td>
<td>Remove a file</td>
</tr>
<tr>
<td>mkdir</td>
<td></td>
<td>Make a directory</td>
</tr>
<tr>
<td>grep</td>
<td></td>
<td>String search utility</td>
</tr>
<tr>
<td>more</td>
<td></td>
<td>Paginate the output to the console</td>
</tr>
<tr>
<td>nano</td>
<td></td>
<td>Simple text editor</td>
</tr>
<tr>
<td>vi</td>
<td></td>
<td>Powerful text editor</td>
</tr>
<tr>
<td>gcc</td>
<td>-o</td>
<td>Compile from source code</td>
</tr>
</tbody>
</table>

**Using Linux (Basic Commands)**

- **dd** Create an image file of a volume or device
- **file** Query a file for its type
- **netstat** List state of TCP/UDP ports
- **dig** DNS Zone transfer
- **host** Look up DNS records
- **lsof** List open files
- **ps** View process list
- **rpcinfo** Enumerate portmapper
- **smbclient** List or use SMB shares
- **md5sum** Calculate MD5 hash

**Administration and Troubleshooting**

- mailsnarf, urlsnarf, filesnarf
- ettercap -q -z MiTM sniffer
- nmap
- hping -c count -S Packet crafter
- snort
- iptables -P -A -j --sport --dport -p Kernel mode firewall
- kismet
- nikto
- maltego
- tcpdump -i Command line sniffer
- firewalk -u Firewall enumerator
- nc -l -e -v "Swiss army knife"
Firewalls and IPTables

The Linux firewall makes a good teaching example because once you understand it, all firewalls are easier. It is free, open source, and widely available.

Types of Firewalls

- **Packet filter**: The simplest form of filtering, looks only at layer 3 and 4
- **Stateful Inspection**: Understands directionality and established sockets
- **Circuit Level Gateway**: Translates sequence numbers along with addresses and ports
- **Application Proxy**: Deep packet inspection all the way into the payload

Attacking Firewalls

- **TCP Flag combinations**: While some flag combinations are filtered, others may pass
- **Firewalking**: Enumerating ACLs on a filter
- **ACK floods**: Overwhelming an SPI firewall into thinking the traffic should pass
- **0th fragment**: Host based firewalls only: The 0th fragment has TCP data, the others do not
- **ICMP redirection**: Hijack local hosts to use the attackers host as a gateway, the traffic can be altered or observed
- **Tunneling and port redirection**: Hiding data inside encapsulation

Setting up a network firewall

A host based firewall only protect the host, a network based firewall must also be a router. In Linux, the Kernel must be told to forward packets:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

There are several default tables for a forwarding firewall to be aware of:

- **INPUT**
- **OUTPUT**
- **FORWARD**
- **ACCEPT**
- **NAT**

IPTables Example: Defending against a Smurf attack

```
iptables -A FORWARD -p tcp -s 0/0 -d x.y.z.m/32 --destination-port 25 --syn -j ACCEPT
iptables -A FORWARD -p tcp -s 0/0 -d x.y.z.w/32 --destination-port 80 --syn -j ACCEPT
iptables -A FORWARD -p tcp -s 0/0 -d x.y.z.w/32 --destination-port 443 --syn -j ACCEPT
iptables -A FORWARD -p tcp -s 0/0 -d 0/0 --destination-port 22 --syn -j ACCEPT
```
IDS and Snort

Intrusion Detection Systems are a key technology for protecting a network. Attackers can also use them to look for very specific events on the network such as logins or other attackers. As a counterpart to firewalls, IDS is a great way to bring together the many of the concepts that been discussed in this course including; sniffing, scanning, and the four major protocols (IP, ICMP, TCP, UDP).

Types of IDS

<table>
<thead>
<tr>
<th>Type</th>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Based</td>
<td>Active</td>
<td>Listens on the hosts</td>
</tr>
<tr>
<td>Network Based</td>
<td>Passive</td>
<td>Listens on the network</td>
</tr>
</tbody>
</table>

Detection Engines

<table>
<thead>
<tr>
<th>Analysis Type</th>
<th>Time</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature Analysis</td>
<td>Real time</td>
<td>Uses a rules based approach</td>
</tr>
<tr>
<td>Anomaly Analysis</td>
<td>Real time</td>
<td>Requires a baseline to compare with</td>
</tr>
<tr>
<td>Statistical Analysis</td>
<td>Not real time</td>
<td>Analysis of patterns and occurrences</td>
</tr>
</tbody>
</table>

Evasion Techniques

<table>
<thead>
<tr>
<th>Technique</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption</td>
<td>IDS cannot decrypt data to look at it</td>
</tr>
<tr>
<td>Fragmentation</td>
<td>IDS might be too busy piecing together traffic and start ignoring some</td>
</tr>
<tr>
<td>Decoy traffic</td>
<td>False positives can confuse investigators</td>
</tr>
</tbody>
</table>

Snort rules

Snort rules take on the following syntax:

```
action protocol address prot -> | <> address prot (option:value; option:value;)
```

Starting Snort

Display layer 2 and 7 to the console, use our own rules file and log here
```
snort -dve -c ./rules.local -l .
```

Examples of Snort rules

The simplest rule
```
alert tcp any any -> any any (msg:"Sample alert"; sid:1000000;)
```

Detecting a simple signature
```
alert tcp 192.168.1.6 any -> 192.168.1.5 139 \\
(msg: “Possible SMBDie Attempt”; content:”|5c 50 49 50 45|”; sid:1000000;)
```

Dynamic rules (May be phased out in favor of a new method called "tagging")
```
activate tcp any any -> any 21 (content:"Login"; activates:1; sid:1000000;)
dynamic tcp any any -> any 21 (activated_by: 1; count:100;)
```
Command Line Tools

The key to becoming comfortable with command line tools is to practice saying in plain language what a command is trying to instruct the computer to do. It's hard to memorize switches and far easier to understand what a tool does. As you study and find more examples, add them to this list.

NMap
nmap -sT -T5 -n -p 1-100 192.168.1.1
Use nmap to run a connect scan at a fast rate without DNS resolution to ports 1-100 at host 192.168.1.1

Netcat
nc -v -z -w 2 192.168.1.1
Use netcat, show on the console a scan that sends packets every 2 seconds to host 192.168.1.1

tcpdump
tcpdump -i eth0 -v -X ip proto 1
Use tcpdump to listen on interface eth0 and display layer 2 and 7 for ICMP traffic

snort
snort -vde -c my.rules -l .
Use snort and show on the console layer 2 and 7 data using configuration file my.rules and log in this directory.

hping
hping3 -I eth0 -c 10 -a 2.2.2.2 -t 100 192.168.3.6
Use hping3 on eth0 and send 10 packets spoofing 2.2.2.2 and a TTL of 100 to host 192.168.3.6

iptables
iptables –A FORWARD –j ACCEPT –p tcp --dport 80
Use iptables and append the forward table with a rule that will jump to the accept table when tcp traffic that has a destination port of 80 is noticed.
Syntax Recognition

The CEH exam requires that you can recognize what an attack looks like from a log file. The following are examples that can be used to help explain the principles of each type of attack:

**Directory Traversal**
http://www.example.com/scripts/../../../../../winnt/system32/cmd.exe?c+dir+c:

**XSS (Cross Site Scripting)**
http://www.example.com/pages/form.asp?foo=%%Cscript%3Ealert("Hacked")%3C/script%3E

**SQL Injection**
http://www.example.com/pages/form.asp?foo=blah'+or+1==1--
http://www.example.com/pages/form.asp?foo=%27%3B+insert+into+usertable+{"something"}

**Nimda Virus**
http://www.example.com/MSADC/../../../../../winnt/system32/cmd.exe?c+dir+c:

**Code Red**
GET/default.ida?NNNNNNNNNN%u9090%u688%u8b00%u0000%u00=a HTTP/1.0

**SNMP OID**
1.1.1.0.2.3.1.2.4.1.5.3.0.1

**Buffer overflow attempt**
Apr 5 02:02:09 [3432] : nops: 62.32.54.123:3211 -> 192.168.3.4:135
0x90/0x90/0x90/0x90/0x90/0x90/0x90/0x90/0x90/

**Zone Transfer**
Apr 5 02:02:09 [3432] : AXFR: 143.32.4.129:4865 -> 192.168.3.4:53

**Enumerate email accounts**
Apr 5 02:02:09 [3432] : VRFY: 78.34.65.45:5674 -> 192.168.3.4:25

**Snort Signature Rule**
Alert tcp any any -> any any (msg:"Test Rule"; sid:1000000;)

**IPTables Rule**
iptables -A FORWARD -j ACCEPT -p udp --dport 53

**Capture Filter**
host 192.168.1.1 and host 192.168.1.2 ip proto 1

**Display Filter**
ip.addr == 192.168.1.1 && tcp.flags == 0x29
Random Recall Exercise

Memorizing a list of tool names is difficult and not actually very beneficial. A better approach is to strengthen your mind’s ability to “think” it has seen all of these things before and map them to an important concept.

The list below is made up of names of tools and malware code divided into groups of five. Sometimes they are related and other times have nothing in common at all. Glance at a group and jot down the first word or phrase that comes to mind and move on to the next group. So not try to explain every item; just one word or phrase an keep going. One term may remind you of something, but your subconscious will see the others as well. On each pass, try to recall something different.

DOS  
Smurf  
SYN flood  
Fraggle  
Buffer Overflow  

Ping OF Death  
Tear drop  
The UNnamed Attack  
Land  
SMB Die  

Chargen  
CPU Hog  
Dos Attack Tools  
Jolt2  
Bubonic  

Land and LaTierra  
Targa  
Blast20  
Nemesys  
Panther2 (Nuke)  

ICMP Packets Sender  
Some Trouble  
UDPFlod  
FSMax  
Trinoo  

TFN (tribe Flow Network)  
Stacheldrach  
TFN2K  
Shaft  
Mstream  

Trinity  
Knight  
KaiTeN  
Worms  
Slammer  

Bots  
Bot Nets
Agobot/Phatbot/Forbot.Xtrembot
SDBot/RBot/UrXBot
mIRC-based Bots-GT-Bots:

DSNX Bots
Q8 Bots
Kaiten
r1-based bots
nslookup

whois
Sam Spade
Smart Whois
NetScan
GTWhois

Xwhois
ARIN
LACNIC
APNIC
DNS Enumerator

subdomain retrieval
Spiderfoot
Domain footprinting tool
SensePost Footprint
Footprinting toolset

Bile
Bile-Weigh
TLD
vet-IPRange
qtrace

vet-mx
jarf-rev
jarf-dnsbrute
Teleport Pro
Wikto

HTTrack Web Copier
Tifny
Google
Google Earth
ciseek.com

DMOZ
Internal URL guessing
Archive.org
Neotrace
VisualRoute Trace

Smart Whois
Email Tacker Pro
Website Watcher (change notification)
GEO Spider
GEOwhere (news search)

Email Spider
Necrosoft Advanced DIG
IANA (Internet Assigned Numbers Authority
3D Traceroute
Kartoo Search Engine

Touchgraph Visual Browser
VisualRoute Mail Tracker
ReadNotify.com (email tracking)
Web Ripper
Robots.txt

Email Spiders
Web Data Extractor
1st Email Address Spider
Power Email Collector Tool
HPing2

Firewalk
Nmap
Blaster Scan
Port Scan Plus
Strobe

IPSecScan
NetScan Tools Pro
WUPS - UDP Scanner
SuperScan
IPScanner

MegaPing
Global Network Inventory
Net Tools Suite Pack
FloppyScan
PhoneSweep - War Dialing Tool

THC Scan
Sandtrap Tool
pof-Banner Grabbing Tool
Httprint Banner Grabbing Tool
Xprobe2

Ring V2
Netcraft URL site
IIS Lockdown Tool
Servermask
PageXchange

Bidiblah Automated Scanner
Qualys Web Based Scanner
SAINT
ISS Security Scanner
Nessus
GFI Languard
SATAN
Retina
Nikto
SAFEsuite Internet Scanner

IdentTCPScan
Cheops
Friendly Printer
Free Proxy Servers (page 352)
SocksChain

Proxy Workbench
Proxymanager Tool
Super Proxy Helper Tool
Happy Browser Tool
Multiproxy

Tor Proxy Chaining Software
Proxy Finder
Proxybag
Proxy Scanner Server
Cheron

Anonymizers
Primedious Anonymizer
Anonymous Surfing Browzar
Torpark Browser
G-Zapper

SSL Proxy Tool
HTTP-Tunnel
HTTP Port
Despoof Tool
What It Is

Sentry PC
Enumeration
SNMP Enumeration Countermeasures
Windows 2000 DNS Zone transfer
Identifying Win2000 Accounts

Active Directory Enumeration
SNMP Enumeration
SNMPUtil
NetBios Null Sessions
NetBIOS Enumeration

DumpSec
NAT
IP Network Browser
User2SID
SID2User

Enum
UserInfo
GetAcct
NewSID
NetBrute

wmidump
ShareEnum
WinFingerprint Utility
snmpenum
wininfo

w2k Active Directory Attack
IP-Tools
getaacct
netview
superscan

enum
pstools
ps exe
ps file
psgetrid

pskill
psinfo
pslist
pslogged on
pspaaswd

psservice
solarwinds
snscan
getif
Network View

The Dude Sniffer
Ethereal
tcpdump
ARP Spoof
Ethercap

Macof
Etherflood
IRS
ARPWorks
Nemesis

arpspoof
dnsspoof
dsniff
filesnarf
mailsnarf

msgsnarf
sshmitm
tcpkill
tcpnice
PhoneSpy
Webmitm
TCP Relay
Effetech
Password Sniffer

MSN Sniffer
SmartSniff
Netwitness
Cain and Abel
Packet Crafter

SMAC
NetSetMan
RAW SNIFFING TOOLS:
Sniffit
Aldebaran

Hunt
NGSSniff
Ntop
pf
iPTraf

EtherApe
Snort
Windump/tcpdump
Etherpeek
Mac Changer

Iris
NetIntercept
WinDNSSpoof
Netfilter
Network Probe

MaaTec Network Analyzer
Antisniff
ArpWatch
PromiScan
AntiSniff

Prodetect
Apple II Virus 1981
Brain 1983
Virdem 1986
Lehigh Virus

IBM Christmas Worm
MacMag
Scores Virus
Internet Worm
AIDS Trojan
VX BBS
Little Black Book (AT&T Attack)
Tequila (first Polymorphic virus)
Michelangelo
DAME (Dark Avenger Mutation Engine)

VCL (Virus Creation Laboratory)
Boza (Windows 95)
Laroux (Excel Macro)
Staog (Excel Macro)
Strange Brew (Java based)

Back Orifice (first remote admin control)
Melissa (Word macro virus and worm)
Corner (ms project)
Tristate (multi-program macro)
Bubbleboy (opening email spread)

Love Letter (fast, shuts down email)
Timofonica (VBS on phones)
Liberty (for PDA’s)
Pirus (PHP scripting)
Gnuman (masked in file sharing)

Winux virus (infects both Windows and Linux)
LogoLogic-A Worm (MIRC chat and email)
PeachyPDF (Adobe PDF worm)
Apple Script worm
Nimda

LFM-926 (against shockwave flash)
Donut (against .net)
Sharp A
Javascript Worm/SQLSpider (MS SQL)
Benjamin (P2P)

Perrun Virus (Jpeg)
Scalper Worm (FreeBSD and Apache)
Sobig (SMTP)
Slammer worm (MS SQL servers)
Lovegate (trojan and worm)

Fizzer (email and P2P)
Welchia
Trojan.Xombe
Randex
Bizex

Witty
MP3Concept
Sassar
Mac OS X
W64.Rugrat.3344

Symb/Cabir-A
JS/Scob-A
WCE/Duts-A
W32/Amus-A
WinCE/Brador-A

JPEG Weakness
SH/Renepo-a
Bofra/IFrame
Santy
MYDOOM

I Love you virus (VBS Script)
Virus Hoaxes
CT Cookie Spy
Dictionary Maker
LophCrack (LC4)

Brutus
AuthForce
Cain&Abel
Munga Bunga
ReadCookies.html

WinSSLMiM
GammaProg
John the Ripper
Obiwan
Hydra

Webcracker
Passlist
Snadboy
RAR
Messenpass

Wireless WEP Key Password Spy
RockXP
PasswordSpectator
Instant Source
wget

Web Sleuth
Black Widow
Window Bomb
Burp
cURL

sitescope Tool
WSDigger
CookieDigger
SSLDigger
SiteDigger

dotDefender
Google Hacking Database (GHDB)
Acunetix Webscanner
Appscan
AccessDiver

Xsite Scripting
SQL Inject
CMD Inject
Cookies/Session Poisoning
Parameter/Form Tampering

Buffer Overflow
Directory Traversal/Forceful Browsing
Cryptographic Interception
Authentication Hijack
Log Tampering

Error Msg Intercept attack
Obfuscation Application
Platform Exploits
DMZ Protocol Attacks
Security Management Exploits

Web Services Attack
Zero Day Attacks
Network Access Attacks
TCP Fragmentation
Log Analyzer

CleanIISlog
Metasploit Framework
Immunity Canvas Professional
Core Impact
UpdateExpert

qfecheck
HFNetchk
cacls.exe
Whisker
N-Stealth HTTP Vul Scanner

WebInspect
Shadow Security Scanner
SecureIIS
Buffer Overflow
$DATA IIS vulnerability

ShowCode.ASP
IIS Directory Traversal
ISSxploit.exe
Msw3prt IPP Vulnerability
WebDav/ntdll.dll Vul

RPC DCOM
ASN exploits
ASP Trojan
URL Poisoning
SQL Injection
Authorization bypass
SQL injection using single quotes
execute OS command
Bad login and bad product list
Getting Output of SLQ Query.

Get Data from DB using ODBC Error message
AutoMagic SQL
Absinthe
SQLDict
sqlExec

SQLbf
SQLSmack
SQL2.exe
AppDetective
Database Scanner

SQLPoke
NGSSQuirreL
SWLPing v2.2
Walking
Wardriving

WarFlying
WarChalking
Blue jacking
GPS
Rogue AP

Fake AP
NetStumbler
MiniStumbler
AiroPeek
WEPCrack, AirSnort

KisMAC
Kismet
WepLab
Wellenreiter
Fatajack

Redfang 2.5
THC-WarDrive
PrismStumbler
MacStumbler
Mognet

WaveStumbler
StumbVerter
AP Scanner
SSID Sniff
Wavemon

Wireless Security Auditor
AirFraf
Wifi Finder
AirMagnet
NAI Wireless

Ethereal
VPNmonitorl
Aerosolve.65
VxSniffer
EtherPEG

DriftNeit
WinDump
Ssidsniff
NetChaser v1.0
WinPcap

AirPcap
BSD-Airtools
AirDefense Guard
WIDZ
Netbios Auditing Tool

Smbbr
SMBCrack Tool
Legion
L0phtCrack
PWdump

RainbowCrack
KerbCrack
NBTDeputy
NetBios Dos Attack
John the Ripper

ScoopLM
SMBRelay
SMBCapture
SMBProxy
SMBGrind

SMBDie
Syskey Utility
Active Password Changer
X.EXE
PsExec

Remoxec
Alchemy Remote Executor
SC-KEylog
SC-Keylog PRO
SpyTestor FTP Keylogger

IKS Software Invisible Keylogger
Ghost Keylogger
KeyGhost USB Keylogger
Perfect Keylogger
Stealth Email Redirector
Spyware
Spector Pro
RemoteSpy
eBlaster

Stealth Voice Recorder
Stealth Keylogger
Stealth Website Logger
Digi-Watcher Video Surveillance
Desktop Spy Screen Capture Program

Telephone Spy
Print Monitor Spy Tool
Wiretap Professional
FlexiSpy
PC Phonehome

Rootkits
Blacklight
Rootkit Revealer
AFX Rootkit 2005
Nuclear

Vanquish
Rootkit Countermeasures
Pathfinder
Rootkit Revealer
Back Orifice

Deep Throat
NetBus
Whack-a-mole
NetBus 2
Girl Friend

Sub Seven
WinTrinoo
Tini
icmd
netcat

Beast
MoSucker Trojan
Proxy Server Trojan
SARS Trojan
Wrappers

RemoteByMAil
HTTP RAT
Shttpd Trojan
Nuclear RAT
BadLucj Destructive Trojan

ICMP Tunneling
ScreenSaver Password Hack
Phatbot
Amitis
Senna Spy

QAZ
Cyber Spy
Subroot Telnet
RECUB
Loki

Sockets de Troie
Masters Paradise
DEvil
Evil
Doly Trojan

Chargen
Stealth Spy Phaze
NetBIOS datagram
ICQ Trojan
MStream

The PRayer 1.0-2.0
Online KEyLogger
Portal of Doom
Senna Spy
Trojan Cow

netstat
fport
TCPview
CurrPorts Tool
Process Viewer

Device Drivers
Registry
Autoruns
Startup List
Tripwire (SIV)

SIV / SFV
MD5sum
ipchains
SARA
gcc

make
chroot
nessus
nmap
cheops

portsentry
iptables
netcat
snort
saint

tcpdump
ethereal
dsniff
hpingle
sniffit

dsniff
lsof
iptraf
lids
hunt

tcp wrappers
LKM
chkrootkit
ntop
lsat

IDS
firewall
honeypot
ids techniques
SIV

sidestep
Tripwire
fragroute
firewall types
firewalk

banner grabbing
HTTP Tunnel
loki
specter
honeyd

KFSSensor