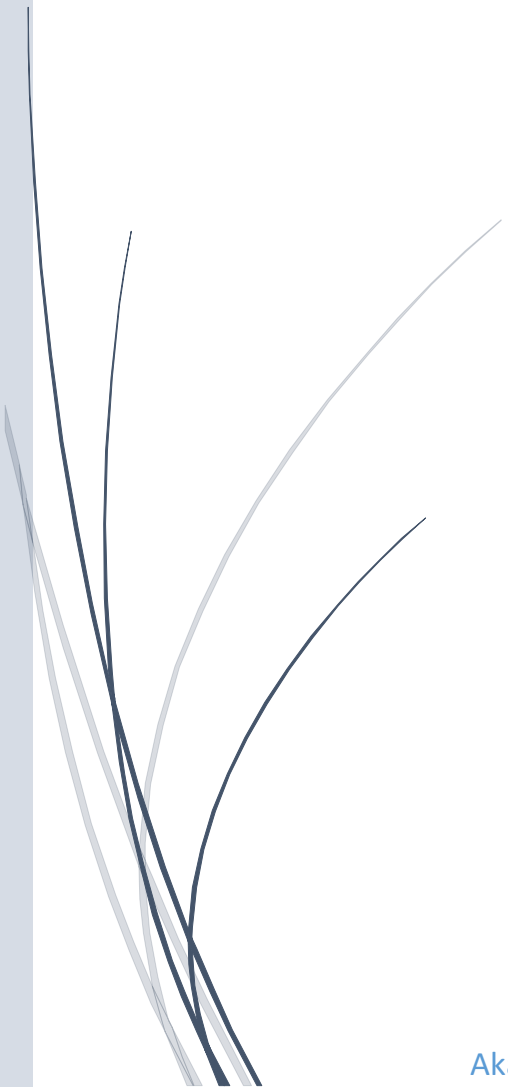
Monday, January 18, 2016

AUTHENTICATION AND COMPUTER SECURITY

Assignment 1



Akash Payne (ap567)

Table of Contents

Question 1 ()

How can attackers bypass intrusion detection systems? Describe in enough technical detail at least 4 ideas, with available tools and countermeasures.

Intrusion detection systems (IDS) are a usually mounted to aid the security provided by firewalls. IDS have weak point, i.e. flaws that are exploited by an attacker to bypass the system: IDS solutions are not perfect every time and IDS administrator are not faultless either. Attackers can try to avoid an IDS they determine the flaws in the design and challenge to exploit these flaws to which each IDS has different strengths and weaknesses.

Network Intrusion detection systems (NIDS) are typically aimed as pass or intrusive monitors of network traffic. This allows single NIDS application to guard a excessive amount of systems as long as they are on the same network segment. A NIDS can detect attacks through one of two methods, either a **signature matching or abnormality detection**. Evasion of abnormality detection NIDS is more luck than skill because the malicious traffic cannot exceed the irregularity thresholds established for the network being observed. Signature matching works similar to virus scanners: each attack has a signature of how it is carried out. When that signature is highlighted on the network the NIDS generates an alert. Abnormality detection NIDS establishes a baseline of the network traffic that is considered normal (Neural Network based Intrusion Detection Systems' 2014). It then alerts when conditions are not normal: not as common due to the large amount of time needed to establish the foundation of their initial high rate of false positives.

Attackers can fool IDS systems into thinking their attack is legitimate traffic. With techniques like *obfuscation, fragmentation, Encryption, Denial of Service* and *application hijacking*, the attacker is able pass attack packets through the IDS by preventing their detection (Insertion attacks, evasion attacks, denial-of-service attacks and complex attacks).

Obfuscation: the process or tool of handling data in such a way that the IDS signature will not match the packet that is accepted, but the receiving device will still interpret it properly. An example, shown in 'The

Hacker's Handbook (2003) describes: sending a packet encoded differently or adding extraneous null characters or junk insertion, packing, junk and first webmail worm (e.g. Yammaner worm, which was spread around only Yahoo! Email accounts). "./c:\windows\system32\explore.exe", interpreted by IDS as "%2e%2fc:\windows \system32\explore.exe", however, a web server would interpret both as strings under the same under the interpretation rules of HTTP. Popular for NIDS and Web Server Vendors are no longer deceived by this illustration, but the principle is still firm for any interpreter that allows alternate command forms – HIDS NIDS. Kevin Townsend (2006) describes the availability of obfuscator tools, such as Metasploit's VoMM and Web Attacker Toolkit intensify the danger that malicious obfuscated code has drastically. IDS developers have started using a real-time event correlation and response approach to counteract more complicated techniques, one such example is DCO. There are even services, such as, FinJan (www.finjan.com) that is offering policy based real-time content inspection in their secure web gateway products. This service will inspect contents of inbound/outbound web traffic regardless of the source and then detect and decode any obfuscated code and dissect HTML code to its components while scanning active components using any sub-engines. Finally, the IDS system will compare the behaviour profile against a list of security protocols and profiles (Finjan, 2009).

Fragmentation: attack breaking an attack into multiple packets (Insertion, Evasion and Denial of Service: eluding Network Intrusion Detection, 1998). Similar to session splicing (divides the string across several packets). Fragmentation of packets occurs normally and hosts are prepared to manipulate receiving data in multiple fragments and potentially in an incorrect order. E.g. attack packet "data" is broken into several different packets, the host at the receiving end will reassemble the fragmented packets and receive the data payload in the fragmented order and reconstruct into the correct arrangement using unique packet sequence number assigned to each packet. A NIDS is only going to see the parts, each part is not an attack packet so the NIDS will not alert anyone. Some NID's will reassemble packets to avoid the fragmentation attacks, but not all NIDS have the processing overhead available to reassemble fragmented packets. Even if the NIDS that can reassemble packets it is physically limited by how many it can or will reassemble. So an attacker can send a fragmented attack and at the same time send a large amount of fragmented placebo packets. While the IDS is attempting to reassemble all the "junk" packets, the fragment attack may be able to go unnoticed by the IDS and If the attack does not have the resources to flood the NIDS the attacker can attempt to wait out the capture buffer on the NIDS. The NIDS will get the first three packets but an entire attack signature since the last part of the fragment is not received in the appropriate time interval, the first three are dropped by the time the last packet gets to the IDS and therefore the host. All packets will be reassembled on the host in an effective attack, but the NIDS won't send an alert as illustrated by EC-Council's book on Ethical Hacking (2009). A widely available tool that can be used is Aircrack-ng's Fragmentation Attack (www.aircrack-ng.org), which is a suite of tools to assess WiFi network security.

Encryption: A NIDS needs to be able to inspect the payload of packets which cross paths to be efficient, this can be malicious in several ways: being encrypted network traffic, when SSL, SSH and IPSec encrypted

tunnels are made: they prevent the NIDS from being able to interpret the packet's true payload, allowing an attacker to use a target's security against themselves. This can particularly be harmful when a system has the same root directory for both http and https, unencrypted and encrypted, web sites. In example, an attacker uses any attack against the HTTPS website, such as SQL injection, buffer overflows and directory traversals, that would work on the HTTP site since HTTPS uses SSL to provide a protected network connection, the traffic is encrypted and consequently it flows past the NIDS without raising an alarm. Equally, encryption helps an attacker avoid being noticed as they do other malicious things after they have compromised the host. There is a rise in popularity of SSL VPNs that will also contribute to this problem.

SSL VPNs: designed to allow portable, easy setup encrypted sessions between client stations and the corporate network. This allows individuals to use public internet terminals to securely connect to corporate private networks to access e-mails and other internal services. The SSL VPN has two challenges for our NIDS: firstly, with an SSL VPN an attacker can attack the network and due to the traffic that is encrypted the NIDS will not be able to detect the attack, even if detected, the chances of that individual being caught are very small. With rising popularity of free internet access provided by restaurants, shops and libraries and a tool like SSL VPN's attackers can attack from anonymous locations with very little fear of being tracked electronically or even being caught. Secondly, even if an IDS were able to decrypt traffic while processing packets to report attacks, the attacker can easily setup a large number of encrypted sessions from other hosts they control and potentially prevent a NIDS from being able to decrypt the traffic, i.e. multiple attack technique. Fred Cohen (1997) discloses that to protect against an attacker encrypting their commands, future NIDS could send an alert if they see an outbound encrypted session from any host that doesn't usually perform encrypted sessions and equally important, any large number of inbound sessions initiations that would be typical of a brute force password attack over SSH or SSL VPN.

Denial of service: to elude NIDS is simple by overloading the NIDS, this can be done many ways: flood the entity with attacks from spoofed IP addresses, creating so many alarms that the security personnel would have a hard chance of finding the actual attacker or attack. This method does depend on the security professional not pulling the plug (expecting such an attack), but to avoid this the attacker could use common attacks that the professional should be aware of and the system is patched against so that there is not any apparent danger. NIDS should differentiate between minor and major attacks, so that the situation is prevented.

Another method is to flood the NIDS with traffic so that it cannot possibly look at every packet and concurrently slip the attack packets past the overloaded NIDS. To be effective, a NIDS must be able to compare the packet data with the signatures of attacks on every packet it inspects, this can prove very difficult for higher speed links (above 100Mbps). Most NIDS vendors have as a minimum one device that they claim about 600Mbps of performance, but these speed solutions are costly. Moreover, NIDS able to process at these speeds would require an enormously large quantity of coordinated, compromised zombie hosts to flood these NIDS thus reducing the likelihood of such an attack as studied by the Information

Security Room (2003). Host-Based Intrusion Detection System (HIDS) analyses each system's behaviour. The HIDS can be installed on any system and are more versatile than NIDS. A host-based system is a program that operates on a system and accepts application and operating system audit logs.

Tools like Low Orbit Ion Canon are freely available tools on the internet allowing attackers to perform denial of service attacks on small servers by sending UDP, TCP or HTTP requests to a victim's server with only an URL of the IP address linking to the server.

Some steps that are performed after detecting an attack are to configure the firewall to filter out the Attackers IP address, play a WAV file that gets the attention of the administrator. Send an SNMP trap diagram to a management console. Send a Windows NT event to the event log. Send an event to the UNIX syslog event system. Send e-mail to an administrator to further notify of the attack. Save the attack information, so time, attacker's IP address, target's IP address or port and protocol information. Save the information of the raw packets for future analysis. Launch separate program to handle the event and forge a TCP FIN/RST packet to force a connection to terminate.

Question 2 ()

How can attackers bypass firewalls? Describe at least 4 possibilities providing enough technical details and some tools and countermeasures, if applicable.

Firewall is a software package or hardware device that defends the resources of a private network from users on other networks. The firewall is a guard around the network that allows only authenticated users to access network resources. This restricts unauthenticated users from entering the network by rejecting them access. Most businesses rely heavily on firewalls for internal security, but there are still many ways for attackers to bypass them. EC-Council (2009) describe firewalls to have limitations just as any current intrusion detection mechanism, for example, they:

- cannot prevent individual users with modems from dialling into or even out of the network, avoiding the firewall completely;
- cannot stop an attacker from getting a user to divulge sensitive information through social engineering (employees must be trained to keep confidential information secret);
- cannot secure against tunnelling4efforts (applications that are secure can be injected with a Trojan tunnelling packets over HTTP, SMTP and other protocol).

Backdoor Tools are programs that Trojan designers use to generate a backdoor into a system for fast, unhindered access. A backdoor tool can be used to remotely access a system for another time, and gives an efficient backdoor allowing an attacker to keep access to infiltrated machines, even though the system administrator identifies the intrusion. Occupytheweb, a user on www.wonderhowto.com, showed that, "resetting password, changing disk access permissions, or fixing original security holes may not be an efficient resolution at all times." *Rootkit* is a collection of tools that is used by intruders to hide his

presence in an attacked system. Another tool is Firekiller 2000 that was designed to be used with other applications and disables firewall and antivirus software, however, with new patches to Antiviruses, such as, McAfee and Norton Antivirus software, these vulnerabilities are easily overcome with up-to-date Antivirus software.

One of the first steps to **counterattacking Trojans** is to uphold a constant upkeep and assessing of basic security policy rules and requirements, having an up to date anti-virus software installed is a good method to protect your machine. However, it won't entirely protect your system, another good method is to consistently at any modifications of programs to ascertain new, odd services or processes that are running. Administration scripts are very useful tool in this regard, particularly when supervising multiple systems. Moreover, a monthly host scan of the network, if there is any chance that an open port at a computer, giving a check as to whether they are authorised. There are many network, application diagnosis and troubleshooting programs, such as *TCPview*, that provide an efficient service, Available at: *https://technet.microsoft.com/en-us/sysinternals/tcpview.aspx* (2011). These tools provide a means to highlight applications opening the port and additionally, avoid using *Netstat* in case of a supplanted or infected file. An example tool of this is *ListDlls* that is used to find any suspicious signs of a Trojan infected or back-doored processes (available at: *https://technet.microsoft.com/en-gb/sysinternals/bb896656.aspx*).

Another method that attackers can use is **port scanning**, which is used to examine ports that victims use: *Nmap* is a very popular tool available or technique used to identify open ports and services available on a network host. **Firewalking** is alternative method used to collect data about remote networks that are behind a firewall. Firewalk probes ACLs on packet-filtering routers and firewalls and is the best recognised software used for this method and involves three hosts:

- firewalking host, the system outside the target network (that the packets are sent to the destination host to access sensitive information);
- gateway host, the system on the target network that is connected to the Internet (through which packets pass on their way to the target network);
- destination host, the target system on the target network that the packets are addressed to (EC-Council, 2009).

Hiding behind a covert channel is called **tunnelling** and is used to bypass a firewall, such as *Loki*, which the concept allows attackers to send Trojan packets by concealing them as normal ICMP_ECHO traffic – as network devices usually do not filter the contents. (Roth 2016) *Loki* can also be used as a way of surreptitiously retrieve information from a system. Detection can be difficult as the only indication is a surplus of ICMP_ECHOREPLY packets with misconstrued payload, so if an attacker keeps traffic on the channel down, they will be able to hide the Loki server inside the kernel and chances of detecting Loki are smaller. The only real countermeasure that could be taken is to disallow ICMP_ECHO and UDP port 53

DNS traffic entirely (UDP port 53 traffic could restrict to the internal DNS server and designate external DNS forwarder and therefore ICMP_ECHO traffic can only be accepted from trusted hosts).

A firewall, as explained by Whitaker, A. and Newman, D. P. (2005), generally "restricts a Trojan client from connecting to a Trojan server. *ICMP tunnelling* and protocol tunnelling show that a firewall can be penetrated, however, firewalls would become impenetrable if countermeasures such as blocking all ICMP traffic were adopted". Other tunnelling methods used *ACK tunnelling* if ICMP is blocked on a firewall, this concept is similar to ICMP tunneling, in the sense that the backdoor application is tunnelling within allowed TCP packets with the ACK bit set. Using the ACK bit as acknowledge receipt of a packet. WindowSecurity (2013) explain that some firewalls and IDS devices do not check packets with the ACK bit set as ACK bits are supposed to be used in response to genuine traffic that has accepted through. AckCmd is a tool that implements ACK tunnelling and is a backdoor application that allows you to use a remote shell on a windows computing that has AckCmd is running on the target host. This allows a client component of AckCmd to communicate with a server component completely through ACK segments.

Moreover, *HTTP tunnelling* is a method that targets a public web server with the TCP port 80 (used for HTTP traffic and is unfiltered on its firewall). This method is a server/client application, the server needs to be uploaded onto the target's system and will tell you which port to redirect through TCP port 80.

A **honeypot** is a countermeasure or method used to attract and trap attackers that are trying to gain access to a system, however, they aren't designed to address any security problems and used mostly as research tools for decoys and to gain information. There are low level interaction honeypots that rely on the emulation of programs that could be found on a vulnerable system. If attacked, the system detects the activity and throws an alert to an administrator. interaction honeypots are more complicated and usually monitor an entire network. Any activity that happens in High level this environment is reported, the main difference being in the setup that handle real systems and real applications as explained by Sean-Philip Oriyano, Jason McDowell (2014).

Question 3 ()

Write an in-depth description of one of the POODLE/Heartbleed/Shellshock vulnerabilities against SSL/TLS, extracting possible security lessons from them and detailing how they have been stopped.

POODLE (*Padding Oracle On Downgraded Legacy Encryption*) is a vulnerability in the design of the encryption standard of **SSL** (*standard secure socket layer*) **version 3.0** that makes the protocol virtually impossible to user securely. The concerns route from design vulnerability in how *SSL 3.0* handles the block cipher mode padding and could be used to steal secretive information from users, such as passwords and cookies. Used by websites and web browser, the vulnerability allows encrypted data to be exposed by an attacker with network access and allows attackers to gain access to a user's private account information

on any website that runs or is running SSL 3.0, i.e. “attacks use POODLE to exploit vulnerabilities to decrypt and pull information from within encrypted transactions” (Stosh, B. 2014).

Heartbleed, an OpenSSL bug, allows an attacker to access information from a client or server’s memory. The information leaks include: private encryption keys, usernames and passwords. Don Parker (28 Aug. 2007) describes that the cost in terms of time and resources to apply such a fix is substantial, not to mention the disruption it causes to business operations. Public internet access is all that Heartbleed and shellshock attackers need to a vulnerable environment in order to launch an attack and exploit a system, simply sending an HTTP request is sufficient to gain information.

Shellshock is a collection of vulnerabilities in bash (the Bourne Again Shell) and is present on almost every Unix-based computer and the use of it allows attackers to take control of a device and/or run programs covertly in the background. It is a common shell for assessing and executing commands from users and other programs. Shellshock allows complete compromise of server environment, while Heartbleed and POODLE can expose sensitive data. With Heartbleed, this data relates both to client and server (e.g. private keys). POODLE mostly exposes client data, such as credit card numbers entered into an online shopping purchase form.

POODLE poses the largest mitigation test of the three, while Heartbleed and shellshock require a simple system fix, there is no real POODLE patch. Mitigating it requires completely removing SSL v3. The impact of doing so is relatively minor for users on web browsers, however, it is enormous for developer’s API implementations, scripted clients, and all sorts of .Net and Java libraries that rely on SSL 3.0, similarly as there a new modified versions of the original POODLE. However, Stosh, B. (2014) argues that while Shellshock and Heartbleed vulnerabilities could be exploited by someone on the other side of the world, for an attacker to effectively exploit POODLE, they would need to be physically close to their victim. Again, this is a relatively easy fix. You can simply instruct your browser not to support the SSL 3.0 standard and set the lower encryption standard to TLS 1.0, which is much more secure. The problem of course is that you won't be able to visit the websites which continue to use SSL 3.0, luckily the is list of websites using SSL 3.0 is getting smaller and smaller.

The POODLE attack takes advantage of TLS clients that implement a downgrade to implement communicate with earlier protocol versions. This downgrade can be triggered by network glitches, or by an attacker. If the attacker can manipulate the network between the client and the server interferes with any attempted handshake offering, for example, TLS 1.0 or later, such clients will readily reveals themselves to the SSL 3.0. as SSL 3.0 and uses either RC4 stream cipher or a block cipher in CBC mode, meaning that there are some biases. An example of this is RC4 biases, meaning that if the same secret, password or cookie, is sent over the over many connections and encrypted with many RC4 streams, more and more information will be leaked. CBC encryption, assuming that an attacker can modify network transmissions between the client and server, there is no real workaround and therefore the only countermeasure would be to avoid SSL 3.0 entirely. Furthermore, Don Parker (28 Aug. 2007) shows that

the most problematic issue with CBC encryption in SSL 3.0 is that the block cipher padding is not deterministic and is not covered by the message authentication code (MAC), therefore the integrity of padding is not fully verified when decrypting.

In web settings is it possible for a man-in-the-middle attacker to decrypt "safe" HTTP cookies, using techniques from a BEAST attack to launch POODLE, run JavaScript agents to get the victim's browser to send cookie-bearing HTTP requests and then intercept and modify the SSL records sent by the browser in a way that the site will accept the modified record, then an attacker can decrypt one byte of the cookies. This allows an attacker to control both request path and the request body and induce requests and will enable them to slowly reveal cookies.

If an attacker was able to access to a system, either by *session hijacking* via POODLE, *straight shell access* via Shellshock or *credential theft* via Heartbleed. The attacker would have unrestrained access to the system, however if that system was running *Bit9* in default-deny enforcement mode, the attacker would be unable to run command-and-control malware to retain persistence on the system because that malware would not be approved to run as discussed by Parker Higgins (January 3, 2015). Moreover, Joe Toomey (2014) articulates that if you were running Carbon Black, the attacker's attempts to execute these processes would result in watch list alerts, and all of their activities (from the moment of their arrival on your system) would be instantly available for analysis and response by your security team. So they got in, but they couldn't run what they wanted to run, and you caught them and kicked them out before they could steal anything.

Question 4 ()

Write an in-depth description of the FREAK SSL/TLS Vulnerability, describing its potential impact and countermeasures/mitigation techniques used.

FREAK (Factoring Attack on RSA-EXPORT Keys) or Factoring RSA Export Keys is a security exploit of in the SSL/TLS protocols. It allows an attacker to intercept HTTPS connections between vulnerable clients and servers and force them to use a weakened encryption that an attacker can break to access sensitive data. Enabling intelligence agencies and now attackers to force clients to use an older or weaker encryption, known as the export-grade key or 512-bit RSA keys.

A vulnerable browser is subject to attack when it connects to a susceptible web server, i.e. a server that accepts "export-grade" encryption, typically RSA_EXPORT cipher suites, as cracking a 512-bit key back can be done a few hours and inexpensive for a single website.

A server can disable support for TLS export cipher suite and other cipher suites that are known to be insecure and enable forward secrecy and also testing the server against Qualys SSL labs or another SSL Server Test tool showed by Thierry Zoller (2009) in his study of possible MITM attacks and discloses that browsers, e.g. Mozilla, Chrome, it is essential to have the most up-to-date version.

A developer can ensure that TLS libraries are unpatched and also up-to-date - OpenSSL, Microsoft Schannel and Apple Secure Transport all suffer from the vulnerability, so these should be avoided until amended. “It is also vital to make sure that software use does not offer export cipher suites, even as a last resort, since they can be exploited even if the TLS library is patched” (Pierluigi Paganini, 2015).

Additionally, Google revealed that an Android patch has already been distributed to partners, meanwhile, they are also calling on all websites to disable support for export certificates and equally, Apple have responded to the FREAK vulnerability and released a statement saying they have a fix in iOS and OS X that are available, IT PRO (2015).

Question 5 ()

Describe in detail all known attacks against the Diffie-Hellman protocol, and the most common countermeasures to stop them.

The Diffie-Hellman (DH) key exchange is a simple public-key algorithm that allows two users to establish a private key using a public-key scheme based on discrete logarithms and is only secure if authenticity of two participants can be established, i.e. a method of communicating public information to obtain a mutual secret and DH is not an encryption algorithm. DH key exchange has the following important properties according to Jean-Fraɳois Raymond and Anton Stiglic (2000):

- The subsequent mutual secret cannot be calculated by either of the parties without the cooperation of the further third party; a third party observing the messages that are transmitted during DH key exchange cannot deduce the resulting shared secret at the end of the protocol.
- The safety of the DH exchange, while being “moderately simple to calculate exponentials modulo a prime, it is very difficult to calculate discrete logarithms”. For large primes, the latter task is considered infeasible.
- DH key agreement is very often used as “part of security protocols or security standards to secure data over public and communication systems, thus the security of the Diffie-Hellman is critical because any weaknesses can lead such systems to become vulnerable against attacks”.

Generic attacks treat any cyclic group as finitely presented, e.g. shanks baby-step giant step, silver-pohlig-hellman, λ-method and Pollard’s ρ method. Special attacks are not generic because they need more information than that provided by the discrete logarithm problem, e.g. they depend on the particular family of groups that exponentiation is employed for that particular cryptosystem.

RSA Laboratories (3.6.1 What is Diffie-Hellman, 2008), illustrates that the DH exchange is vulnerable to a man-in-the-middle attack. In this attack, an opponent intercepts a victim’s public value and its own public value to a target. When it transmits its public value, the opponent substitutes it with its own and sends it to the victim. The opponent and the victim thus agree on one shared key and then the opponent and the target agree on another shared key. After this exchange, the opponent simply decrypts any messages sent

out by the victim or the target, and then reads and possibly modifies them before re-encrypting with the appropriate key and transmitting them to the other party, an example of a tool is, the Man-in-the-Middle Attack Framework, which provide a modular and easily modifiable framework.

This vulnerability is present because DH exchange does not authenticate the participants. Possible solutions include the use of digital signatures and other protocol variants. The authenticated Diffie-Hellman key agreement protocol or Station-to-Station (STS) protocol was developed to defeat this vulnerability, the immunity achieved by allowing the two parties to authenticate themselves to each other by the use of digital signatures and public certificates.

Bibliography

→ Aitel, D. and Young, S. (2003) The Hacker's Handbook: The Strategy Behind Breaking into and Defending Networks: Auerbach Publications.

→ Bodo Möller, Thai Duong, Krzysztof Kotowicz (September 2014) This POODLE Bites: Exploiting The SSL 3.0 Fallback, https://www.openssl.org/~bodo/ssl-poodle.pdf: OpenSSL.

→ Don Parker (2 April 2008) Analyzing a Hack from A to Z (Part 4), Available at: http://www.windowsecurity.com/articles-tutorials/misc_network_security/Analyzing-Hack-Part4.html (Accessed: January 18th 2016).

→ Don Parker (28 Aug. 2007) Virtual realms and rootkits, Available at: http://www.windowsecurity.com/blogs/parker/security-central/virtual-realms-and-rootkits-198.html (Accessed: January 18th 2016).

→ EC-Council (2009) Ethical Hacking and Countermeasures: Secure Network Infrastructures, Course Technology.

→ Finjan (2009) Security Policies In-Depth, Vital Security 9.2. [Online]. Available from: https://www3.trustwave.com/software/secure_web_gateway/manuals/9.2.0/Security%20Policies%20In-Depth-9.2. (Accessed 20th January 2016).

→ Fred Cohen (1997) 50 Ways to defeat your Intrusion Detection System, fred at all.net: Fred Cohen & Associates.

→ Information Security Room (2003) Intrusion detection evasion: How Attackers get past the burglar alarm, Chicago: SANS Institute.

→ IT PRO (10 Mar, 2015) Don't FREAK out over the Factoring Attack on RSA-EXPORT Keys, Available at: http://www.itsecurity.com/features/news-feature-metasploit-vomm-102906/ (Accessed: January 19th 2016).

→ Jean-Franc¸ois Raymondm Anton Stiglic (2000) Security Issues in the Diffie-Hellman Key Agreement Protocol, http://instantlogic.net/publications/DiffieHellman.pdf: Zero-Knowledge Systems Inc.

→ Joe Toomey (15 October 2014) Bit9 + Carbon Black Poodle (SSLv3 Vulnerability) Status, Available at: https://blog.bit9.com/2014/10/15/bit9-carbon-black-poodle-sslv3-vulnerablity-status/ (Accessed: January 18th 2016).

→ Kevin Townsend. There's a new kid on the block, going by eVade o' Matic Module, or VoMM for short, Available at: http://www.itsecurity.com/features/news-feature-metasploit-vomm-102906/ (Accessed: January 19th 2016).

→ occupytheweb (2014) Hack Like a Pro: How to Hijack Software Updates to Install a Rootkit for Backdoor Access, Available at: http://null-byte.wonderhowto.com/how-to/hack-like-pro-hijack-software-updates-install-rootkit-for-backdoor-access-0149225/(Accessed: 10th January 2016).

→ Parker Higgins (January 3, 2015) Three Vulnerabilities That Rocked the Online Security World: 2014 in Review, Available at: https://www.eff.org/deeplinks/2014/12/three-vulnerabilities-rocked-online-security-world-2014-review (Accessed: January 16th 2016).

→ Patcek, T.H. and Newsham, T.N. (1998) Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection, http://insecure.org/stf/secnet_ids/secnet_ids.html: Insecure.org.

→ Pierluigi Paganini (March 12, 2015) The FREAK Vulnerability: From Discovery to Mitigation, Available at: http://resources.infosecinstitute.com/the-freak-vulnerability-from-discovery-to-mitigation/ (Accessed: January 17th 2016).

→ Roth, F (2016) LOKI- Indicators of Compromise Scanner, Available at:http://www.darknet.org.uk/2016/01/loki-indicators-compromise-scanner/ (Accessed: 16th January 2016).

→ Russinovich, M (2011) ListDLLs v3.1, Available at: https://technet.microsoft.com/en-gb/sysinternals/bb896656.aspx (Accessed: 12th January 2016).

→ Russinovich, M (2011) TCPView v3.05, Available at: https://technet.microsoft.com/en-us/sysinternals/tcpview.aspx (Accessed: 11th January 2016).

→ Sean-Philip Oriyano, Jason McDowell (2014) CEH: Certified Ethical Hacker Version 8 Study Guide, 8 edn., Canada: John Wiley & Sons.

→ RSA Laboratories (2016) *3.6.1 WHAT IS DIFFIE-HELLMAN?* Available at: *http://webcache.googleusercontent.com/search?q=cache:YVXFpt22Dm0J:uk.emc.com/emc-plus/rsa-labs/standards-initiatives/what-is-diffie-hellman.htm&num=1&hl=en&gl=uk&strip=0&vwsrc=1* (Accessed: 16th January 2016).

→ Sodiya A.S., Ojesanmi O.A., Akinola O.C., et al. (2014) ' Neural Network based Intrusion Detection Systems', International Journal of Computer Applications, 106(18), pp. 19-24 [Online]. Available at:http://research.ijcaonline.org/volume106/number18/pxc3899636.pdf (Accessed: 10th January 2016).

→ Stosh, B. (2014) Ouch! Nasty POODLE Variant Bypasses TLS Crypto Affecting Over 10 Percent of the Web, Available at: https://freedomhacker.net/nasty-poodle-variant-bypasses-tls-crypto-hitting-major-sites-3506/ (Accessed: 21/01/2016).

→ Thierry Zoller (December, 2009) New SSLv3 / TLS vulnerability - MITM attacks possible, Available at: http://blog.zoller.lu/2009/11/new-sslv3-tls-vulnerability-mitm.html(Accessed: January 16th 2016).

→ Whitaker, A. and Newman, D. P. (2005) Penetration Testing and Network Defence, Cisco Press.

→ WindowSecurity (2013) Backdoors, Available at: http://www.windowsecurity.com/whitepapers/unix_security/Backdoors.html (Accessed: 21st January 2016).