**RIS 420 - Research Assignment #1**

Members Names: Thomas Reid Zuk, Cyril V, Mike Nguyen

Date of submission: March 6th, 2016

**Table of Contents**

Netbios Page 3

Microsoft-ds Page

DNS Page

IRC Page

FTP Page

LDAP Page

RADIUS Page

MySQL Page

**NetBIOS 137-139:**

**Description:**

NetBIOS stands for network basic input/output system. NetBIOS is an API that operates on the Session layer of the OSI model to provide a communication interface between an application and a network. It provides name resolution, datagram and session services across 3 ports; 137, 138 and 139 respectively. NetBIOS packets can be transported over many different Network layer protocols such as TCP/IP, UDP and IPX. Data using protocol 139, the session service, utilizes TCP, while ports 138 and 137 services utilize UDP. From RFC 1088 " NetBIOS is a standard which specifies a means of creating virtual circuits and of transmitting and receiving point-to-point, multicast, and broadcast datagrams."[1]. NetBIOS allows for old systems and applications to operate on a TCP/IP network. According to RFC 1002(1987) NetBIOS allows for various node types to be accommodated on local and internet topologies with or without IP Broadcasting. NetBIOS achieves this by encapsulating IP PDUs within a NetBIOS PDU and assigning IP numbers to hosts. A business or organization would use NetBIOS services in a device that is not supportive of modern IP network capabilities for supporting legacy services and devices. Most uses of NetBIOS are older file transfer services and file servers.

**Vulnerabilities:**

If not properly mitigated NetBIOS over TCP/IP can provide large amounts of information on a network such as hosts, services running and network layout. The Session service provided by the NetBIOS over TCP/IP allows two devices to establish a connection, having this exposed could allow someone to intercept and manipulate these services.

Using a tool called NBSTAT NetBIOS services on port 139 can reveal large amounts of information about a host. Computer name, remote name, IP Address, list of local NetBIOS names, contents of the session table and destination session IP addresses.

**Risks and Risk Mitigation:**

An attacker gaining such detailed information about a network and system they could use it to find other vulnerabilities within a network or a host. With the information gained using NBSTAT an attacker could possibly connect to a device if null sessions are allowed. From there they would be able to see all the connected shares. An attacker could retrieve sensitive data that is shared on the network which could have various levels of impact against the business such as financial reports, business plans, product planning and any other information employees are sharing through the service.

To help mitigate the risks the following steps should be taken. Disable null sessions, have strong administration passwords, have strong passwords for file shares if they are in place, disable the Guest account, do not allow root access remotely, do not share system folders such as Windows folders.

**How information is collected/logging:**

Since NetBIOS is a transport protocol and not a software or standalone service is does not keep logs. A device running a service that utilizes NetBIOS may keep logs like a file server may keep track of how connected, downloaded, uploaded files, but this is not logging NetBIOS itself.

**Wireshark Analysis:**

[1]https://tools.ietf.org/html/rfc1088

**Microsoft-DS: 445**

**Description:**

Microsoft-DS port 445 stands for Microsoft Directory Services. It is a Microsoft Active Directory service that is required for Active Directory access and authentication. (msdn site) It has been implemented for use since the launch of Windows 2000. Port 445 and Microsoft-DS replaced the usage of port 139 in NetBIOS service for file sharing and transfer. Port 445 is known as "SMB over IP". Business would use Microsoft-DS for file sharing services in Windows environments either locally, across the internet or through a VPN that connects multiple locations together. RFC's that were found about microsoft-ds were limited and did not include any relevant information to how the protocol operates or the services it served. The only information in the RFC's found were a list of services names and transport protocol port number that it uses.

**Vulnerabilities:**

Leaving port 445 open has been known to leave system vulnerable to worms which can have any malicious intent designed by the creator. A list of some past vulnerabilities related to leaving port 445 open are Sasser Worm, Korgo AB, HLLW Moega, Trojan Netdepix, Windows null session exploits and various root kits. One famous example is the Iraqi worm or iraqi\_oil.exe which uses a combination or port 445 open, weak passwords and privilege escalation to gain access to a target machine.

NetBIOS worms using port 445 have used the PSExec tool and other tools to replicate by continually scanning networks and the internet for other machines with port 445 open and replicating themselves. A trojan called Lioten is associated with port 445 vulnerabilities.

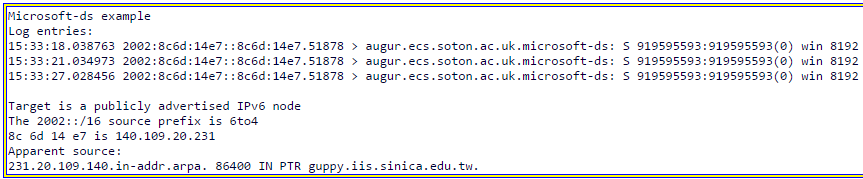
**Risks and Risk Mitigation:**

Remote control of a Windows machine with port 445 is possible. If port 445 is open on a host that host is highly susceptible to an attack and can even have a worm uploaded to the system. Users stand to have their machine remote controlled which means any documents may be stolen, any passwords or accounts that the machine has access to may be compromised and their machine may be used in further attacks to further spread the malicious code.

Many ISP have taken risk mitigation of port 445 vulnerabilities into their own hands by blocking any traffic using port 445. Although many ISP's help to prevent attacks associated with port 445 it is not a good idea to leave it solely in their hands. On a router blocking port 445 traffic will help to ensure no traffic uses port 445. In general disabling or blocking any machine exposed to the internet with port 445 is the best way to protect against an attacker.

**Information gathered and Logging:**

Microsoft-DS stores log entries of devices that connect to a device running Microsoft Active Directory services. Below is an image of a Microsoft-ds log file.



[1]

https://www.ietf.org/proceedings/68/slides/v6ops-6/sld7.htm

**Wireshark Analysis:**

**DNS: 53**

**Description:**

DNS stands for Domain Name System and is a vital backbone for the functionality of the internet. DNS resolves domain names such as google.com and facebook.com into IP Addresses. All internet devices has a IP address, so DNS is a critical service for all users of the internet, especially non-technical users. Ports above 49152 can be used to receive local DNS and Remote DNS queries and responses to a local or remote DNS server. From RFC 1035 "From the user's point of view, domain names are useful as arguments to a local agent, called a resolver, which retrieves information associated with the domain name."[1] DNS can be encrypted which adds a level of security for the user knowing that their DNS is not being tampered with by a man-in-the-middle attack.

All websites, businesses, charities and personal websites alike all user DNS servers in some way or another. Large businesses like Google.com have their own DNS server that both the public and internal operations can use. Most websites use an external DNS provider when they purchase a domain name for their website to manage domain name to IP resolution services. Consumers looking to make personal websites may use a site such as GoDaddy.com to buy a domain, who in turn manages DNS services. Within an organizations network an internal DNS may be used to resolve names of local servers to make services easier to use such file servers.

https://www.ietf.org/rfc/rfc1035.txt

**Vulnerabilities:**

Due to the nature of how many users browse the internet, through domain names, exploiting DNS services is a large target for attackers. DNS spoofing and DNS cache poisoning are two methods of exploiting DNS services that expose users to many attacks. Users can be exposed to man-in-the-middle attacks where-by they make a DNS request and at some point in the request cycle an attacker tampers with the request or response to redirect someone to a malicious website.

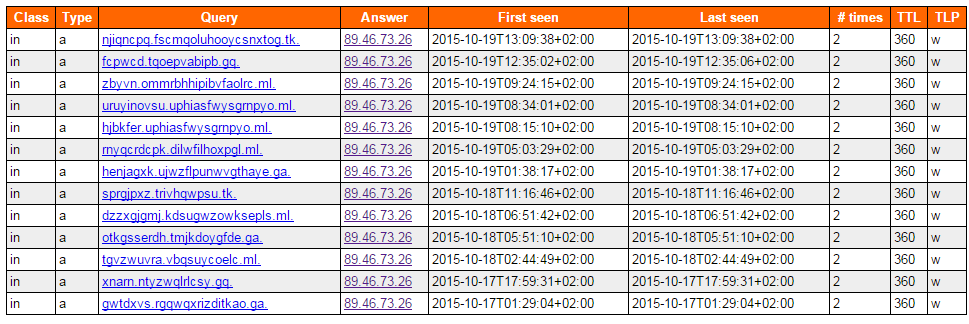
**Risks and Risk Mitigation:**

A device connected to a spoofed DNS could have any traffic querying a domain name redirected to a malicious site, traffic intercept (man in the middle) and traffic manipulated. A possible outcome of a DNS attack could be a user requests access to their banking website, and an attacker alter the request or the response to point to a phishing website they have designed to look like the banking website. The unsuspecting user would enter their log in information and it is now in the hands of the attacker. From there a smart attacker would forward the credentials to the bank website and redirect the user so they don't have any suspicion that they were just compromised.

A good way to prevent a device from being exposed to a DNS spoofing attack is to manually configure the DNS settings of a device to use only approved and properly configured DNS servers such as Google DNS. For organizations with services that use internal domain names a local DNS is required. Configuring machines to use the internal DNS as the primary DNS and then using a trusted DNS for internet queries is a great way to reduce and eliminate DNS spoofing attacks. Services called DNSCrypt are also available which encrypts DNS traffic so someone performing a man-in-the-middle attack could not tamper with DNS traffic, or make it exceedingly challenging to tamper with. DNSSEC is also a service available for DNS security. DNSSEC provides authentication as well as a chain of trust to establish a confidence in the response of DNS queries.

**Gathered information and Logging:**

Logging of DNS on both the DNS server and the clients that utilize a DNS can be turned on and various level of details of logging can be specified. Below is a image of a sample DNS log.



https://www.invincea.com/wp-content/uploads/2015/10/T-Online-malvertising-attack-DNS-logs-image-1.png

**Wireshark Analysis:**

**IRC: 194**

**Description:**

IRC stands for Internet Relay Chat, it is used for exactly what the name implies, for text communication over the internet. There are many web application and client side IRC clients that are popular internet chat rooms for various use and well known for use by hacker groups such as LulzSec and Anonymous. Many IRC networks use alternative ports instead of 194, such as 6667, 6697, and 994. Some web site based IRC networks list they have used the follow ports 6665, 6666, 6667, 8000, 8001, 8002, 6697, 7000, 7070. Some ports are associated with SSL/encrypted traffic, others are associated with plain text IRC networks.

IRC is a text chat service, so any text sent using IRC services could be expose. This could include website links, personal information, stories, and general conversation information. The most common use for IRC by a business is running an IRC service for users. IRC is likely not used for internal business operations as it is unprofessional and very casual in natural. Most businesses would elect to use more secure and professional services like email.

Many browser based IRC services use SSL to encrypt their users traffic. Freenode, a popular IRC client uses 6697, 7000, 7070 ports for SSL traffic.

https://tools.ietf.org/html/rfc2813

**Vulnerabilities:**

Some IRC services that are browser based inherit vulnerabilities such as cross site scripting, remote code execution, expose users IP's which have lead to DDOS and DOS attacks. Privilege escalation of the chat room or chat channel has been noted in some clients and services using IRC but this does not compromise a users machine. With privilege escalation a ordinary user could gain admin access of the channel and lock down the channel, kick users and prevent users from talking. IRC botnets do exist and were prime in the early years of the internet during the 1990's and 2000's and at relatively scarce today.

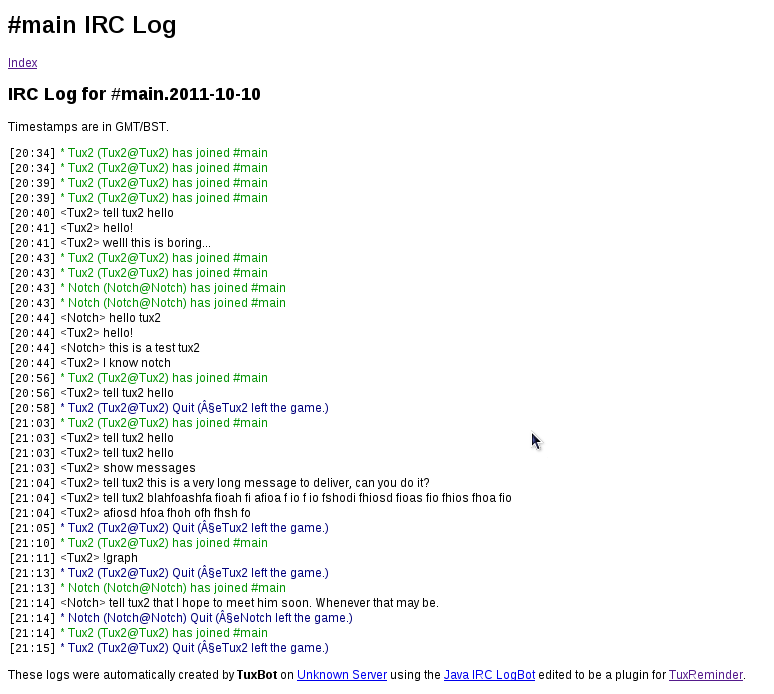
**Risks and Risk Mitigation:**

Any information shared across an IRC chat room, if it is not encrypted and not password protected can be spied on. Before sharing and private communications over IRC ensure the traffic is encrypted, the channel is private, or no users have joined that you do not want to share your communication with. There are well known standalone and browser IRC clients and information on their security can be found easily. An example of a standalone IRC client is mIRC, an example of a popular browser based client is Freenode IRC.

**Gathered information and logging:**

Both the server and client of IRC can log the chat and is decided by the user whether they store their chat logs and up to the administrators of the server if they want to log the chat or not. Logging could store basic information on the client side like names of the users, the channel name, the server address/name, users communications and time stamps of related events. The IRC server could log all this information and in addition log a users IP address.

Sample IRC log



http://dev.bukkit.org/media/images/34/653/irc-log.png

**Wireshark Analysis:**

**FTP: 20-21**

**Description:**

FTP stands for File Transfer Protocol, which allows for a client to be able to send or receives files from a server. FTP operates in Application Layer of the TCP/IP suite and OSI Models. The File Transfer Protocol was originally published in April 16 1971 as RFC 114, and originally ran on the model that predated the current TCP/IP suite. The updated version became RFC 765 in June 1980 which was when it migrated to TCP/IP, and further updated to its current documentation in October 1985 as RFC 959. Although FTP is most known to operate on port 20 as this is used for the actual data transfer, it also utilizes port 21 to as a control in which is used for communication between the client and server. FTP will use the TCP protocol in the transport layer in order to ensure that files successfully reach their destination. This data transfer has three possible modes which are stream mode, block mode and compressed mode. These are self-explanatory but FTP additionally has two modes of activity, passive and active which determine how the protocol establishes a connection between client and server. Passive mode was added in September 1998 in RFC 2428 alongside of support for IPv6. Other updates to FTP in the form of security extensions came in RFC 2228 which was published in October 1997. This was implemented to enhance confidentiality and integrity for the protocol and prevent sensitive information such as passwords from being viewed. It also protects commands, replies and data transfer between the client and server. Logs for this protocol are primarily kept in /var/log in Linux.

**Vulnerabilities**

FTP is not designed as a secure protocol and in RFC 2577, there is a list of issues and vulnerabilities that are associated with FTP. In the document is list that FTP is susceptible to a bounce attack. This attack involves sending the FTP “port” command to an FTP server containing the network address and port of the machine and service which is being attacked. This allows the attacker to attack another machine using the FTP server to push a file to that host through a vulnerable service. Also FTP is very susceptible to brute force if not properly configured. However the most easily exploited issue in FTP is that its traffic is not encrypted. This means that anyone doing packet capture will be easily able to capture the plaintext that is being sent to and from the FTP server. FTP is also vulnerable from port stealing. Lastly ‘by making a legitimate transfer, an attacker can observe the port number allocated by the server and guess the next one that will be allocated. This can allow the attacker to use all available ports and deny the service or allow the attacker to steal a file meant for someone else.’[1]

**Risk and Mitigation**

Since FTP was designed before the time of encryption, we find it to have many problems such as the possibility of exposing information due to the plaintext. Also the bounce attack allows for attackers to learn more about other machines that interact with the FTP server and learn what ports and services it has open/up. It will take extra precautionary steps in order to strengthen FTP and bring it up to standard. It is ideal to set-up a limit to the amount of attempts one can try to login to the server. This will lessen the possibility of being vulnerable to a brute force attack. RFC 2228 also lists security extensions that will better secure communication between server and host and make it not plaintext. It will also hide commands that are being used during the communication. In order to deal with the FTP bounce attack, it is stated in RFC 2577 to not permit “port” commands to ports whose TCP numbers are less than 1024 ( well-known ports utilized by many computers). This will still leave ports higher than 1024 which tend to be less important vulnerable but this is ideally good enough precaution. A final possibility is to disable the “port” command entirely if you want to totally mitigate the command being abused. FTP servers should also implement rules to limit which network addresses has the ability to access and download its files. It should be strictly only available to specified networks and remote hosts. Standard FTP will issue a response when the USER command which is used to ask whether this is a valid username exists. If it doesn’t exist, a response of 530 will appear otherwise if it does a response of 331. It is ideal that the setup always return 331, so that it will not be possible to continually guess at valid usernames. To avoid the possibly of port stealing, the FTP server should use random local ports for communication. There are additional security extensions in RFC 2428 for FTP usage with IPv6.

There are now other alternatives, or rather derivatives of the FTP protocol such as FTPS and SSH FTP which comes with more security features. FTPS utilizes Transport Layer Security (TLS) and Secure Sockets Layer (SSL) while SSH FTP uses Secure Shell Protocol (SSH) to connect and transfer files.

**Wireshark Analysis**

[1] https://tools.ietf.org/html/rfc2577

https://en.wikipedia.org/wiki/File\_Transfer\_Protocol

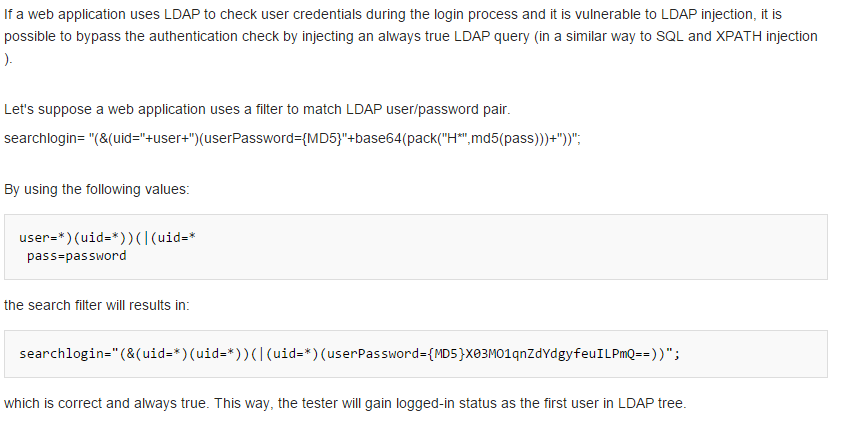
**LDAP: 389, 636, 3268**

**Description:**

LDAP stands for Lightweight Directory Access Protocol. LDAP is an open and vendor free service which means that it is not owned by any major company. As the name implies it is used for Directory information services and contains information about its users, computers, networks, and services and applications. It is ideally used to keep all information in a single location. LDAP was first introduced in RFC 1777 in March 1995. At the time it was meant to work with the pre-existing X.500 Directory system but require less resources than the Directory Access Protocol (DAP) that was in place. Unlike the DAP system which was built on the OSI model, LDAP was purposed to use the working model of the TCP/IP Suite. LDAP and DAP have numerous different functions, but both follow the same structure. LDAP is able to perform the various tasks such as ADD, BIND, Delete, Search and Compare, and Modify entries in directories and databases. A company implementing LDAP may use it as a place to grab usernames and passwords. Since LDAP is a protocol for accessing directories and accessing databases, it can be used to accomplish a wide variety of tasks that mostly benefit from a centralized repository. With the update to LDAPv3 in 1997 found in RFC 2251, it added TLS (Transport Layer Security) to the protocol. The most recent update found in RFC 4511 provides the most up to date revisions to the protocol. LDAP by default will operate using TCP and UDP port 389. A common way to secure LDAP communication is by using an SSL tunnel, which by default utilizes port 636. LDAP is able to use LDAP-compliant directories, and thus pull its information. An example of this is the Global Catalog (GC). The GC which is a role given to a Domain Controller (DC) is a LDAP-compliant directory which holds a partial representation of every object can be accessed through port 3268. Searches done through the GC will also use this port. OpenLDAP houses its logs in /var/log by default with the name slapd.log.

**Vulnerabilities**

Using LDAP with the TLS connection enabled with the command “StartTLS” with LDAPv3 and up is very safe. However like many database applications such as SQL, LDAP can suffer from a server-side attack known as LDAP-injection. This is an attack which uses improper LDAP statements which could result in the attacker gaining information that should not be available. Similarly to SQL injection, the same principles and techniques apply to LDAP injection. This is a big deal as OWASP still lists SQL injection as a critical web application security risk (top 10). An example of this is found in the image below, and shows how it is possible to break the logically system. LDAP injection has many different angles of attack that may allow the attacker access unauthorized content, evade application restrictions, gather unauthorized information, or add/modify Objects inside the LDAP tree structure.

[2]

**Risk and Mitigation**

LDAP as a centralized repository means that it holds a lot of information regarding company staff, assets, etc. that could become compromised. In order to prevent communication from being captured, it is mandatory to ensure that TLS is running by using the “StartTLS” command. TLS/SSL was an optional part of LDAPv2, but in the current LDAPv3 has been made into a mandatory staple. Alternatively you can just run LDAPS which uses port 636 and automatically runs the protocol in SSL. In RFC2829, they outline the basic threats to the LDAP protocol which are listed below:

1. Unauthorized access to data via data-fetching operations
2. Unauthorized access to reusable client authentication information by monitoring others' access
3. Unauthorized access to data by monitoring others' access
4. Unauthorized modification of data
5. Unauthorized modification of configuration
6. Unauthorized or excessive use of resources (denial of service)
7. Spoofing of directory: Tricking a client into believing that information came from the directory when in fact it did not, either by modifying data in transit or misdirecting the client's connection.

[3]

These issues can be solved by implementing client and server authentication using SASL (Simple Authentication and Security Layer) in order to prevent unauthorized access or directory spoofing. TLS/SSL as mentioned above will prevent threat 2, 3 and 7. Additional information can be found in RFC2829. LDAP offers 3 different methods to authenticate users. They are No Authentication, Basic Authentication, and SASL (Simple Authentication and Security Layer). In No Authentication, it is as the name implies and is primarily used for information that is meant to be publically shared and of no value. Basic Authentication utilizes a login system however, this information is either sent in plaintext or base64 encoded. These two methods will not provide enough security for a company. SASL is a framework that enables the use of multiple different types of security mechanisms. These can include Kerberos, TLS, Digest-MD5 and many others. In order to combat the possibility of LDAP-injection, user/client input will need to be sanitized to prevent these dangerous search queries. The use of escape characters to replace existing ‘problem’ characters or symbols is ideal to solve this issue. These are just some ways to secure LDAP and keep sensitive information safe.

**Wireshark Analysis**

https://www.ietf.org/rfc/rfc1777.txt

https://en.wikipedia.org/wiki/Lightweight\_Directory\_Access\_Protocol

[2] https://www.owasp.org/index.php/Testing\_for\_LDAP\_Injection\_(OTG-INPVAL-006)

https://www.owasp.org/index.php/LDAP\_injection

[3] https://www.ietf.org/rfc/rfc2829.txt

**RADIUS: UDP 1645-1646(old), 1812-1813; TCP 1645-1646, 3799, 2083 (Experimental)**

**Description:**

RADIUS stands for Remote Authentication Dial-In Service and is a networking protocol that provides AAA for users who connect and use it. AAA stands for Authentication, Authorization and Accounting and is provided by RADIUS in a centralized form. RADIUS is most typically run on both UNIX and Windows Machines. The details for Authentication and Authorization can be found in RFC2865 while Accounting is found in RFC2866. Developed by Livingston Enterprises Inc. in 1991, RADIUS is a client/server protocol that at the time used UDP ports 1645 (authentication) and 1646 (accounting). At the time, it was a proposed solution to control dial-in access to NSFnet from Merit Network. It was not until 1997 that the protocol was added to the IETF standards in RFC2058. As documented, RADIUS servers are responsible for receiving user connection requests using AAA and sending back configuration information to the client to deliver service. RADIUS is primarily used by internet service providers to access clients thus establishing a connection to the Internet. An access client might be an end-user trying to dial-in to a service provider, or be VPNs and wireless access points. These clients send their credentials in the form of a RADIUS message to a RADIUS server, where this information is authenticated and a reply is sent back. Nowadays RADIUS will use UDP 1812 for authenticating and UDP 1813 for accounting instead of 1645 and 1646 respectively. It is also possible for possible for RADIUS to communicate with TCP instead of UDP. In the experimental RFC 6613 and 6614, RADIUS is sent similarly over TCP port 1812 and 1813. There is also the additional TCP port 3799 for dynamic authorization. Secured RADIUS using TLS (RFC 6614) will default use TCP port 2083. As aforementioned, RADIUS is primarily used by ISPs but it is possible that a company with a large set of IPs can make great use of this also. A UNIX machine with FreeRADIUS has its log configuration in the radius.conf. For Windows you will have to configure the Network Policy Server to perform accounting (logging) for the various features of RADIUS. DIAMETER is meant to be a successor to RADIUS.

**Vulnerabilities**

Since RADIUS still primarily operations using UDP packets, this means that it is possible for an attacker to specially craft packets or forge packets with more ease compared to TCP. This also means that the protocol is more vulnerable to spoofing. Most attacks revolve around the shared secret, which is used by RADIUS in conjunction with MD5 hashing to conceal the password. In the case of a Response Authenticator Attack, if the observer sees a valid Access-Request, Access-Accept or Access–Reject packet sequence, an offline attack can be used to try to crack it. The attacker will have to compute the MD5 for the compiled fields and attempt to recreate the same hash. Other possible avenues of attack may include replay attacks that use replays of server responses, such as the Access-Accept and gain access without any login credentials. RADIUS only protects the user credentials, but other information that is passed through such as tunnel-group or VLAN memberships can be easily accessed. It is also noted that an attacker can perform a Denial of Service attack on a Network Policy Server (NPS) by using a packet with carefully crafted username strings that prevents authentication for the NPS thus preventing it from connecting. The Request Authenticator for RADIUS is not truly random and an attack may able to guess the next possible correct request.

**Risks and Mitigation:**

Since RADIUS deals with enabling connections for the different access clients, it is ideal that the RADIUS message be safe from possible attackers that may be listening. Potential risks that may arise from RADIUS is that like mentioned before, RADIUS by default only secures information regarding the username/password, and other pieces of information is available to prying eyes. By using IPsec, the whole RADIUS message can become encrypted and the RADIUS sensitive fields such as Access-Request, Access-Accept and Access-Reject can be hidden also. The method that RADIUS uses to secure the usernames and passwords with shared secret and MD5 is not ideal. As the aforementioned topic of attacks on the shared secret is quite common for RADIUS one may implement the Radsec protocol found in RFC 6614 to strengthen RADIUS. Radsec claims to fix the flaws in the RADIUS sensitive fields and also strengthens the username and passwords. Since MD5 is proven to be weak and UDP being much easier to spoof for, Radsec adds TLS (Transport Layer Security) Encryption for RADIUS. It also makes it so communications no longer exist on the UDP transport protocol but rather the TCP transport protocol.

**Wireshark Analysis**

http://www.juniper.net/techpubs/software/aaa\_802/sbrc/sbrc70/sw-sbrc-admin/html/Concepts2.html

https://en.wikipedia.org/wiki/RADIUS

https://tools.ietf.org/html/rfc2058

https://tools.ietf.org/html/rfc6613

https://tools.ietf.org/html/rfc6614

http://books.gigatux.nl/mirror/wireless/0321202171/ch13lev1sec4.html

**MySQL: 3306**

**Description:**

MySQL is a relational database system that was created by Oracle in 1995. MySQL has a registered port of 3306, meaning the Internet Assigned Numbers Authority (IANA) has assigned and the port for use with the MySQL port. No other ports are commonly associated with a MySQL data base because it is a registered port and service.

Being a database, MySQL could potentially expose any information stored in a database that uses MySQL. MySQL is one of the top relational database platforms and is used by organizations such as NASA, Verizon, and YouTube. MySQL may be used by any software, website, or web application for its relational database needs. Encrypting MySQL traffic, via a SSL certificate for a website, is highly recommended. Sending unencrypted database traffic exposes any connection to the database to interception of data. This could be anything from price of a product on a eCommerce platform, to credit card information, to usernames and passwords. MySQL on a website would be server side so the actual query to the database would not be sent from a users browser to the server, but the data retrieved and being submitted to a database should be encrypted.

**Vulnerabilities:**

Databases are a large target for many attacks, user data ranging from pointless internet forum comments, to names, passwords, addresses and emails can be very valuable for attackers. MySQL has had its fair share of vulnerabilities in the past, and there are currently no known vulnerabilities. The past year, 2015, saw only 1 vulnerability published and fixed, in comparison to 2014 which saw 38 vulnerabilities. SQL injection is a infamous vulnerability with the SQL platform, MySQL has only had 4 known total SQL injection vulnerabilities, all of which have been patched.

**Risks and Risk Mitigation:**

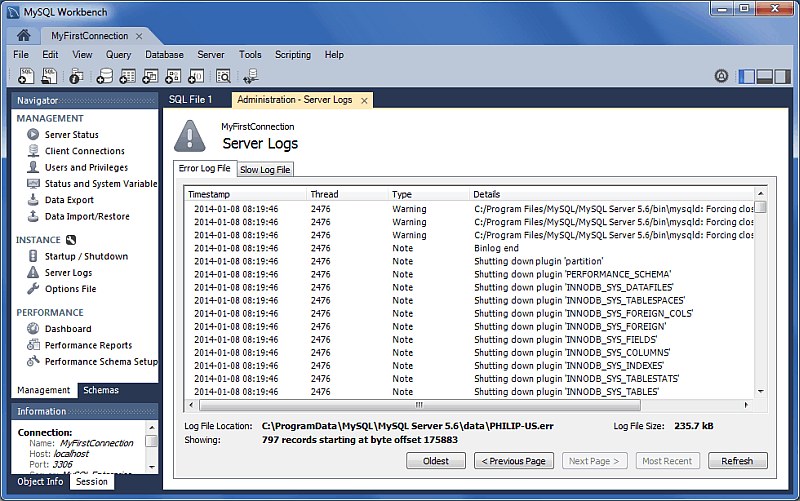
Practices to help prevent any exploitation of MySQL services include never allow 'root' access remotely, never use the 'root' user account for executing queries from a service such as a website. Have strong passwords for all accounts, especially the root account. Keep MySQL services up to date to prevent against any known vulnerabilities that may be discovered.

**Gathered Information and Logging:**

MySQL servers support logging of various types of information the service uses. Logs of queries performed, time to execute query, and statistics can be recorded. Creating statistics of the logs is often done to help improve efficiency of a database. Logging of users connected to the database and the queries they run is highly recommended for security. Knowing which user performed which queries can help trace a possible attack, loss of data and recovering of data.

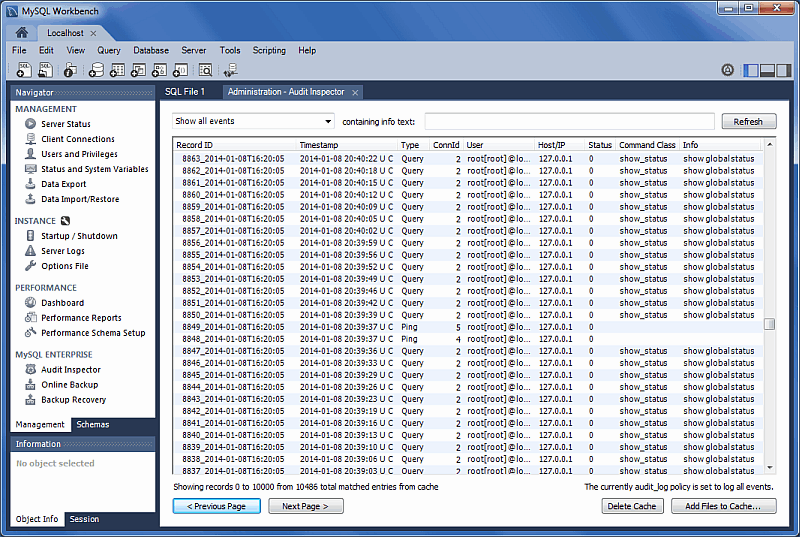
Sample of a MySQL server event log.

https://dev.mysql.com/doc/workbench/en/images/wb-navigator-server-logs-error-log.png



Sample of MySQL server query log

https://dev.mysql.com/doc/workbench/en/images/wb-audit-inspector-main.png



**Wireshark Analysis:**

**Sources**

http://www.techrepublic.com/blog/it-security/the-problem-with-netbios/

http://www.pcmag.com/encyclopedia/term/47773/netbios

https://technet.microsoft.com/en-us/library/cc940063.aspx

https://wiki.wireshark.org/NetBIOS

https://isc.sans.edu/port.html?port=445

http://www.speedguide.net/port.php?port=445

https://msdn.microsoft.com/en-us/library/dd979226(v=crm.6).aspx

https://tools.ietf.org/html/rfc7194

http://www.speedguide.net/port.php?port=194

https://freenode.net/irc\_servers.shtml

https://technet.microsoft.com/en-us/library/dd197515(v=ws.10).aspx

https://developers.google.com/speed/public-dns/

https://ca.godaddy.com/help/what-is-dns-665

https://www.infoblox.com/sites/infobloxcom/files/resources/infoblox-whitepaper-cybercriminal-guide-exploiting-dns.pdf

http://dev.mysql.com/doc/refman/5.7/en/introduction.html

https://www.mysql.com/customers/