Abstract

Database is the heart of most of the web applications, as it stores data necessary for a website and application to "Survive" `and run the business. Undoubtedly, it needs to be guarded well! Yet, even after implementing various security measures, the story of database hacking and data breach is rather conspicuous and extensive. One wonders, why? Probably, evolution of technology with substantial focus on ease of use, enlarged network environment with network based applications and lower skill level needed for manipulations has made online world more insecure and vulnerable to attacks.

The objective of this paper is to present an overview on various types of hacking attacks and its impact on the applications which are dependent on the database, as well as our business. The paper also discusses mitigation strategies and use of tools to conduct penetration test to uncover vulnerabilities in the system and make it secured further and better resistant to hacking attacks.

Keywords: Database and data security, Hacking attacks, Mitigation Strategies, Tools, Best Practices.

INTRODUCTION

The term data refers to qualitative or quantitative attributes of a variable or set of variables. Data (plural of "datum") are typically the results of measurements and can be the basis of graphs, images, or observations of a set of variables. Data are often viewed as the lowest level of abstraction from which information and then knowledge are derived [1]. These derivatives of data are quite significant to us and particularly in organisations since information and knowledge are the cardinal elements for running nearly all the processes efficiently and beneficially. The useful and critical data is generally stored in the relational databases discovery. This data helps in crucial decision-making and is widely used by the organisations to either for further processing or knowledge stay ahead of their competitors. Hence, databases and data are soul of an organisation and their security is of utmost importance.

Technology is growing at a rapid pace and past few years have seen a steep rise in the availability of electronic resources. Technologies such as mobile phones, laptops, PC's, Internet, websites, and extensively networked environment have added another dimension to information technology industry. This new dimension introduces various means at hacker's disposal to commit attacks at multiple numbers of locations. Crimes can occur easily as the pace at which the new technology is developing is far ahead than the pace of making the new technology secured. Internet and websites are now backbone of information exchange system and we can see it in our daily routine/every aspect of life. Both at organisational as well as individual levels we extensively use e-mail services, banking and financial services, social networking sites, online shopping services. This involves exchange of critical information and data on internet and this information can easily be misused for illegal purposes if it is not secured properly.

Information and Data security – the safeguarding of computer systems and the integrity, confidentiality, and availability of the data they contain [2] – has long been recognized as a serious issue. Its importance is growing as computer usage advances in vast aspects of modern life. In addition cyber-attacks or breaches of information security, appear to increase in frequency and, few are willing to ignore the possibility that severity of future attacks could be much worse than what has been observed till date.

This paper presents valuable insight into types of hacking attacks, mitigation strategies to avoid them, a new model of data security and finally a case study demonstrating Penetration testing of XYZ Corporation (name withheld) with the use of the various available tools to conduct the testing process along with really thought-provoking results showing how **successfully running** websites are vulnerable and exploitable.

Last Decade's History of Security Attacks:

1999: The infamous "Melissa" virus infects thousands of computers with alarming speed, causing an estimated $80 million in damage and prompting record sales of anti-virus products [3].

2000: Yahoo, eBay, Amazon, Datek and dozens of other high-profile Web sites are knocked offline for up to several hours following a series of so-called "Distributed denial-of-service attacks"[3].

In June 2005, MasterCard announced that up to 40 million credit card holders were at risk of having their data stolen -- and 200,000 definitely had -- because of a Trojan on the computers of a credit card processing company. [4]

In 2008 Sony lost 101 million accounts due SQL injection attacks.

October 12, 2011 Sony has suffered a data breach involving the usernames and passwords of about 93,000 customers. Attackers were able to reuse to logon to people's PlayStation Network (PSN), or Sony Online Entertainment (SOE), or Sony Entertainment Network (SEN) accounts [4].

The underlying point behind the history is that present security systems though efficient, are not necessarily sufficient to prevent further security attacks. There can be a vaccine for polio but steep hills are yet to climb in fighting autoimmune diseases. If a hacker wants to get inside a system, chances are he/she would find a way and there is not much one can do about it but yes, the one thing that can surely be done is to **make it harder** for them to get in. We have to supplement our traditional ubiquitous cover of firewalls, antivirus softwares and IDPS systems with a well- known methodology, called Penetration Testing. This would **make it somewhat infallible and better resistant to attacks** and reinforce the principle of "Prevention is always better than cure".

Overview: Types of Hacking Attacks

Before proceeding further we need to understand what are the various type of hacking attacks that a system could be open to. Top 10 web security attacks for 2010 are [5]

A1: Injection
A2: Cross-Site Scripting (XSS)
A3: Broken Authentication and Session Management
A4: Insecure Direct Object References
A5: Cross-Site Request Forgery (CSRF)
A6: Security Misconfiguration
A7: Insecure Cryptographic Storage
A8: Failure to Restrict URL Access
A9: Insufficient Transport Layer Protection
A10: Invalidated Redirects and Forwards

Below discussed are 4 major security threats with severe business impact (but rest of them can't be overlooked in any case).

A1: Injections–Injections deceive an application into including unintended commands in the data sent to an interpreter (Hibernate etc.). They are very prevalent, particularly in legacy code, often found in SQL queries, LDAP queries, XPath queries, OS commands, program arguments, etc. Injection flaws are easy to discover when examining code, but more difficult via testing. Scanners and fuzzers can help a penetration tester to find them.
Business Impact-Severe- Injection can result in data loss or corruption, lack of accountability, or denial of access. Injection can sometimes lead to complete host takeover. Reputation of an organisation could be gravely damaged.

A2: Cross-Site Scripting (XSS)- XSS is one the most prevalent web application security flaw. XSS flaws occur when an application includes user supplied data in a page sent to the browser without properly validating or escaping that content. There are three known types of XSS flaws: 1) Stored, 2) Reflected, and 3) DOM based XSS. Detection of most XSS flaws is fairly easy via pen testing or code analysis but web 2.0 technologies, such as AJAX, make XSS tougher to detect via automated tools.
Business Impact-Severe- Results in hijacking user sessions, deface web sites, insert hostile content, redirect users, hijack user's browser using malware. Consider the business value of the affected system and all the data it processes along with the business impact of public exposure of the attack.

A3: Broken Authentication and Session Management- Developers frequently build custom authentication and session management schemes, but building these correctly is a hard job. As a result, these custom schemes frequently have flaws in areas such as logout, password management and timeouts, remember me, secret question, account update, etc. Finding such flaws can sometimes be difficult, as each implementation is unique.
Business Impact-Severe- Such flaws may allow some or even all accounts to be hacked. Once successful, the hacker has access to the application in the same way as the victim.

A7: Insecure Cryptographic Storage- The most common flaw in this area is simply not encrypting

data that deserves encryption. When encryption is employed, unsafe key generation and storage, not rotating keys and weak algorithm usage is common. Use of weak or unsalted hashes to protect passwords is also common. External attackers have difficulty detecting such flaws due to limited access. They usually must exploit something else first to gain the needed access.

Business Impact-failure frequently compromises all data that should have been encrypted. Typically this information includes sensitive data such as health records, credentials, personal data, and credit cards. Consider the business value of the lost data and impact to your reputation. What is your legal liability if this data is exposed? Also consider the damage to your reputation?

Mitigation strategies:

Injection attacks

Firstly test that use of interpreters clearly separates untrusted data from the command or query. To ensure this use bind variable in all prepared statements and stored procedures. Bind variables allow the interpreter to distinguish between code and data. We could also avoid the use of interpreters by use of a safe API's which provides a parameterized interface. If a parameterized API is not available, we should carefully implement escaping mechanism on user inputs using the specific escape syntax for that interpreter [5]. Use of stored procedures typically assists to avoid SQL injection attacks by restricting the types of statements that can be passed to their parameters. Finally make use of SQL Query Parameters. Scrub the input data to make sure it will contain only acceptable characters (a-z, A-Z, 0-9).Dynamic query building should be avoided. Penetration testers can validate injection issues by shaping exploits with an intruders mind that confirm the vulnerability. Automated dynamic scanning which exercises the application may provide insight into whether some exploitable injection flaws exist. In addition to that code reviews, updates on patches to avoid Zero day attacks are absolute necessities.

Cross site scripting

XSS can be easily avoided by keeping untrusted data separate from active browser content. The preferred option is to properly escape all untrusted data based on the HTML context (body, attribute, JavaScript, CSS, or URL). Testers must test for input validations to ensure that all user supplied input sent back to the browser is verified to be safe (via input validation), and that user input properly escapes before it is included in the output page, proper output encoding ensures that such input is always treated as text in the browser, rather than active content that might get executed. Complete coverage requires a combination of manual code review and manual penetration testing. In addition to this consider employing Mozilla's new Content Security Policy that is coming out in Firefox 4 to defend against XSS [5].

Broken Authentication and Session Management

Following strategies are quite useful in avoiding these attacks. A Penetration Tester should ask following questions when testing an application:

- Are credentials always protected when stored using hashing or encryption?
- Can credentials be guessed or overwritten through weak account management functions (e.g., account creation, change password, recover password, weak session IDs)?
- Are session IDs exposed in the URL (e.g., URL rewriting)?
- Are session IDs vulnerable to session fixation attacks?
- Do session IDs timeout and can users log out?
- Are session IDs rotated after successful login?
- Are passwords, session IDs, and other credentials sent only over TLS connections?

Answers to these questions would verify that authentication and session management mechanisms are implemented properly. In addition to that a pen tester could also test for these points:

- Authentication should be simple, centralized, and standardized and always be sure that SSL protects both credentials and session id at all times
- Check your SSL certificate
- Examine all the authentication-related functions
- Verify that logoff actually destroys the session
- Password should be complex with larger combinations of characters
- Authentication error message should be generic and it should not reveal any information
- Methods described to avoid XSS flaws should be used properly as XSS can be used to steal session ID's.

Insecure Cryptographic Storage

- Identify all sensitive data: The first thing to determine is which data is sensitive enough to require encryption. For example, passwords, credit cards, health records, and personal information should be encrypted. For all such data, ensure:
  - It is encrypted everywhere it is stored for long term, particularly in backups of this data.
  - Only authorized users can access decrypted copies of the data
  - A strong standard encryption algorithm is used.
  - A strong key is generated, protected from unauthorized access, and key change is planned for.

Considering the threats you plan to protect this data from various attacks (e.g., insider attack, external user) make sure you encrypt all such data in a manner that defends against these threats

- Ensure threat model accounts for possible attacks
- Use encryption to counter the threats; don't just 'encrypt' the data. Protect data with appropriate mechanisms File encryption, database encryption, data element encryption etc.
- Generate, distribute, and protect keys properly
- Be prepared for key change
- All keys, certificates, and passwords are properly stored and protected
- Safe key distribution and an effective plan for key change are in place
- Analyse encryption code for common flaws

These security measures certainly help a penetration tester and database administrators to protect the data and databases from the malicious users. But to **make it harde**r for hackers to get into the system we should use these 20 critical controls to avoid sophisticated and highly advanced technological attacks [6]

Critical Controls Subject to Automated Collection, Measurement, and Validation:

1. Inventory of Authorized and Unauthorized Devices
2. Inventory of Authorized and Unauthorized Software
3. Secure Configurations for Hardware and Software on Laptops, Workstations, and Servers
4. Secure Configurations for Network Devices such as Firewalls, Routers, and Switches
5. Boundary Defense
6. Maintenance, Monitoring, and Analysis of Security Audit Logs
7. Application Software Security
8. Controlled Use of Administrative Privileges
9. Controlled Access Based on Need to Know
10. Continuous Vulnerability Assessment and Remediation
11. Account Monitoring and Control
12. Malware Defences
13. Limitation and Control of Network Ports, Protocols, and Services
14. Wireless Device Control
15. Data Loss Prevention

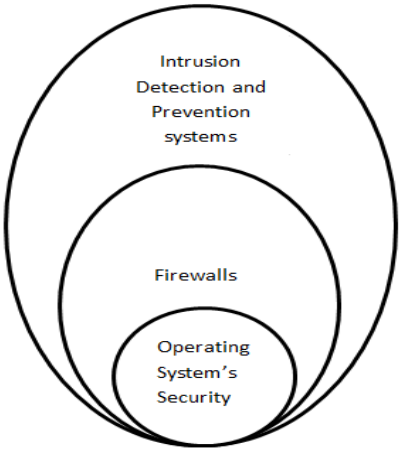
Additional Critical Controls are:

16. Secure Network Engineering
17. Penetration Tests and Red Team Exercises
18. Incident Response Capability
19. Data Recovery Capability
20. Security Skills Assessment and Appropriate Training to Fill Gaps

Here I would like to propose a new model of data security.

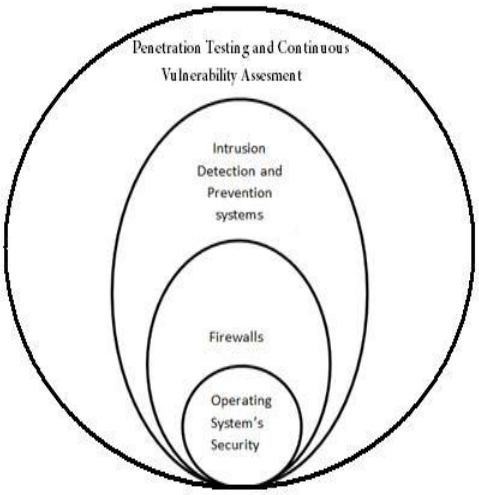
Existing Model:

Figure-1



Proposed Model:

Figure-2



This model could be easily applied on the web as well as on the servers along with standalone applications) in which Penetration Tests and Red Team Exercises should be used as a first line of defence instead of additional control. Main idea behind the model is to use the penetration testing against one of the own system as we are the one who certainly know much more about our own system than malicious users (Hackers and crackers). We also need to focus on the areas where we are lacking to find new arising vulnerabilities in systems due to exponential rate of development of new and existing technologies. Rapid development of technology in IT industry is blessing us with incredible functionalities which are highly user friendly and really easy to use. The "Ease of Use "factor is really critical here as it is decreasing the skills set required by a "Hacker" to penetrate a system and it is alarming as it is decreasing by the day.

Many softwares and systems are released with known and unknown security holes and bugs which may cause vulnerabilities in the future and provide gateways by which threats are manifested. There are also ambiguities in the process of reporting vulnerabilities. Also the process of applying critical patches is slow which gives an easy chance to hackers and black hat groups to exploit vulnerabilities to the fullest until correction patch is applied. If data is hacked and stolen, most organizations – particularly private businesses – have strong incentives not to share information about attacks. CERT estimates that as much as 80% of actual security incidents go unreported, in most cases because the organization was unable to recognize that its systems had been penetrated, or there were no indications of penetration or attack, or the organization was reluctant to publicly admit to being a victim of a computer security breach.[8]
Therefore a prime Principle for the organisations should be **"To Know it First, And Fix it first"** (It could only be ensured by the Penetration Testing), as hackers always works on the principle of **"To Know it first and Exploit it first"**. It's easy to find out how long a gun safe will resist the attention of a thief armed with an acetylene torch. It's impossible to find out how long your server will resist the attentions of a thief armed with hacking tools publicly available on the Internet! It is the race between White hat (Penetration Testers) and Black Hats (Hackers and Crackers). Penetration test of one's own system first to find out bugs, holes and vulnerabilities would certainly give edge and make it hard for them to crack an already Pen tested and secured system. This process is analogous to the vaccination.

Process: Planning and Execution of a Penetration Test [11]

Penetration test should be planned with an intruder's mind and that's why now I use the term hacker instead of penetration tester throughout the process.
There are five basic steps traditionally used by computer hackers to gain unauthorized access, and subsequently take over computer systems. These five steps may be used to plan a computer attack for purposes of cybercrime or cyber espionage. The steps are frequently automated through use of special hacker tools that are freely available to anyone via the Internet. Highly-skilled hackers use automated tools that are highly sophisticated, and their effects are initially quite difficult for computer security staff and technology to detect. These sophisticated hacker tools are usually shared only among an exclusive group of other highly-skilled hacker associates.

Step 1. Reconnaissance
In this first step, hackers employ extensive pre-operative surveillance to find detailed information about an organization that will help them later gain unauthorized access to computer systems. The most common method is social engineering, or tricking an employee into revealing sensitive information (such as a telephone number or a password). Other methods include dumpster diving, or rifling through an organization's trash to find sensitive information (such as floppy disks or important documents that have not been shredded).This step can be automated if the attacker installs on an office computer a virus, worm, or "Spyware" program that performs surveillance and then transmits useful information, such as passwords, back to the attacker. Spyware" is a form of malicious code that is quietly installed on a computer without user knowledge when a user visits a malicious web site. It may remain undetected by firewalls or current anti-virus security products while monitoring keystrokes to record web activity or collect snapshots of screen displays and other restricted information for transmission back to an unknown third party.

Step 2.Scanning
Once in possession of special restricted information, or a few critical phone numbers, an attacker performs additional surveillance by scanning an organization's computer software and network configuration to find possible entry points. This process goes slowly, sometimes lasting months, as the attacker looks for several vulnerable openings into a system network and bypass firewall security. Another way of scanning for vulnerabilities is called "War Driving", where hackers drive randomly through neighbourhood

trying to detect signals from business or home wireless networks. Once a network is detected, the hacker may park nearby and attempt to log on to gain free, unauthorized access [9]. New "Anti forensics tools" are now available on the Internet that allow hackers to more effectively hide their actions, and thus defeat more investigators who search for technical evidence of computer intrusions [10].

Step 3.Gaining Access

Once the attacker has developed an inventory of software and configuration vulnerabilities on a target network, he or she may quietly take over a system and network by using a stolen password to create a phony account, or by exploiting a vulnerability that allows them to install a malicious Trojan Horse, or automatic "bot" that will await further commands sent through the Internet.

Step 4: Maintaining access

Once a hacker has gained unauthorized access, he or she may secretly install extra malicious programs that allow them to return as often as they wish. These programs, known as "Root Kits" or "Back Doors", run unnoticed and can allow an attacker to secretly access a network at will. If the attacker can gain all the special privileges of a system administrator, then the computer or network has been completely taken over, and is "owned" by the attacker. Sometimes the attacker will reconfigure a computer system, or install software patches to close the previous security vulnerabilities just to keep other hackers out.

Step 5: Covering Tracks

Sophisticated attackers desire quiet, unimpeded access to the computer systems and data they take over. They must stay hidden to maintain control and gather more intelligence, or to refine preparations to maximize damage. The "Root Kit" or "Trojan Horse" programs often allow the attacker to modify the log files of the computer system, or to create hidden files to help avoid detection by the legitimate system administrator. Security systems may not detect the unauthorized activities of a careful intruder for a long period of time.

First three steps are very important for testing point of view as moving successfully from first to third step would give a total insight into weakness and vulnerabilities of the system. In step 4 and 5 a penetration tester could test the stability, robustness and integrity of a system.

Tools: Penetration Testing

These tools assess security vulnerabilities in networks or host systems and produce a set of scan results. However, because both administrators and attackers can use the same tool for fixing or exploiting a system, administrators need to conduct a scan and fix problems before an attacker can do the same scan and exploit any vulnerability found. Below are the samples of tools used for penetration testing.

Penetration Testing Frameworks:

1. Metasploit Pro 4.0
2. Linux BackTrack (Penetration Testing Distribution)

Database Scanners

Name	Open source	Proprietary Product	OS/Engine
Scuba by Imperva database Vulnerability scanner		Yes	Various
Shadow database Scanner		Yes	Various
SQLdict	Yes		MS – SQL
ISS Database Scanner		Yes	Oracle, Sybase, SQL

Network Based Scanning Tools

Name	Open Source	Proprietry Product	OS
Cisco Secure Scanner		Yes	NT
Whisker	Yes		Linux
Retina		Yes	NT/2000
Nessus	Yes		Unix/Linux
Nmap Footprinting tool	Yes		Unix/NT

Host Based Scanning Tools

Name	Open source	Proprietary Product	OS/Engine
Microsoft Baseline Security Analyser(MBSA)		Yes	Windows OS
Altiris SecurityExpressions		Yes	-
Tara	Yes		Unix/Linux

Wardialers

Name	Open source	Proprietary Product	OS
Toneloc	Yes		DOS based but run on win 95
THC -SCAN	Yes		Dos.Win 95 /98/NT
Securelogix Telesweep Secure		yes	Windows NT 4.0 and 2000

CASE STUDY

Let us now take case study that will further highlight the benefits of Penetration Testing of a system to make it more secure.

Case Study: To conduct the Penetration Test on a web application under test

Definition: The Penetration testing team primarily should detect the vulnerabilities and holes in the system and then try to exploit them according to the authorization and permissions.

Problem statement: The primary problem with many organizations is that they focus on existing security infrastructure that addresses only the network and server software threats. But the data security capabilities required to be compliant goes far beyond these technologies. Network and server software protections (network firewalls, Intrusion Prevention Systems), while important, provide no insight into data-level attacks targeted directly against a database or indirectly via a web application. As most prevalent attack against the databases are SQL injection attacks and are highly sophisticated so it can be challenging for the penetration tester how to use the generalized concept of penetration testing to test the web applications against the SQL injection?

Solution: Understand that data security is an on-going process. Traditional process of penetration testing could be customized according to the need of the project.
This part of the paper would describe how to pen test the web application against the SQL injection attacks.

Step 1: Initiation
Information is generally provided by the client about the web application to test. But this information is not always sufficient and pen tester should also collect the information about the website. Open the website in Mozilla FireFox as it is supplemented with some basic but good tools to find the SQL injection vulnerabilities, and with good Content security policy released with FireFox 4.0.
Website of XYZ Corporation is tested during the research.

Step 2: Scanning: Activity performed by Tools
Tool: Install SQL Inject Me and run it. It lists all the web pages and input fields including hidden fields (which can be the potential injection points in the applications) of the web application.

Now inject all the input fields with SQL Injection strings using the SQL Inject me, but it is always beneficial to inject various combinations of some common SQL injection strings manually.
Examples of common SQL Injection strings are:

- ‘OR’’=’
- 1 OR 1=1
- 1' OR '1'='1
- 1'1
- 1 EXEC XP_
- 1 AND 1=1
- 1' AND 1=(SELECT COUNT(*) FROM tablenames); --
- 1 AND USER_NAME() = 'dbo'
- '; DESC users; --
- 1' AND non_existant_table = '1
- ' OR username IS NOT NULL OR username = '
- 1ANDASCII(LOWER(SUBSTRING((SELECT TOP 1 name FROM sysobjects WHERE xtype='U'), 1, 1))) > 116
- 1 UNION ALL SELECT 1,2,3,4,5,6,name FROM sysObjects WHERE xtype = 'U' –
- 1 UNI/**/ON SELECT ALL FROM WHERE
- %31%27%20%4F%52%20%27%31%27%3D%27%31
- 1' OR '1'='1
- &#49&#39&#32&#79&#82&#32&#39&#49&#39&#61&#39&#49

Appearance of long error message due to injection od of these strings is the first indication for positive result that site may be vulnerable to attacks.
Error message example: *Unclosed quotation mark after the character string‘)>1 AND ‘=’ and tut_pass=’OR’=’’incorrect Syntax near ‘)>1 AND ‘=’and tut_pass=’OR’=’’* showing website is vulnerable to the SQL injection attacks. Error messages also reveal information like Technology used for the creation of the web site with version number e.g. ASP, ASPX, PHP, HTML with Version.X. This information can be used to find

public vulnerabilities of the specified version to craft more precise penetration test.

Tool 2: For further scanning Acunetix Web Vulnerability Scanner (NFR Evaluation Edition) is used which have given the following results

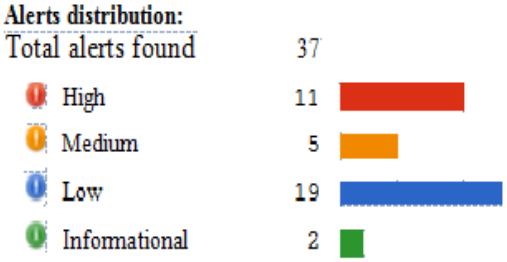
Acunetix Website Audit: Detailed Scan Report-Scan of http://www.XYZ.com:80/

Server Information

Responsive	True
Server banner	Apache/2.2.9 (Fedora)
Server OS	Unix
Server technologies	ASP,ASP.NET,PHP,Perl, Java/J2EE,ColdFusion/Jrun, Python, Ruby, mod_ssl,mod_perl,mod_python, OpenSSL,FrontPage

Threat Level:
Level 3 – High
One or more high-severity type vulnerabilities have been discovered by the scanner. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface your website.

Alerts distribution:



Knowledge base:
Open Ports:
Open Port **21 / ftp**
Port Banner:
Open Port **22 / ssh**
Port Banner:
Open Port **25 / smtp**
No port banner available.
Open Port **80 / http**
Port Banner:
Open Port 3306 / MySQL
Port Banner:
A MySQL server is running on TCP port 3306.

Alerts summary:

Cross Site Scripting

Affects	Variations
/signin.php	4
Severity	High
Type	Validation
Reported by module	Scripting (XSS.script)

URL encoded POST input **txtPassword** was set to **"onmouseover = prompt (967672) bad="**.The input is reflected inside a tag element between double quotes.
Impact: Malicious users may inject JavaScript, VBScript, ActiveX, HTML or Flash into a vulnerable application to fool a user in order to gather data from them and can hijack the session

Recommendations:
Your script should filter meta characters from user input.

POP3 weak password

Severity	**High**
Type	Configuration
Reported by module	Scripting (pop3_bruteforce.script)

The POP3 server is using a weak password. Tool was able to guess the credentials required to access this resource.

Affected items

Server
Details
Username: temp , Password: e0e0e0e0
Server
Details
Username: temp , Password: wampp
Server
Details
Username: temp , Password: test
Server
Details
Username: temp , Password: temp
Server
Details
Username: test , Password: qwerty
Server
Details
Username: test , Password: 12345

Impact: An attacker may access the password protected content
Recommendations: Enforce a strong password policy. Don't permit weak passwords.

Apache httpd Remote Denial of Service:

Severity	**Medium**
Type	Configuration
Reported by module	Scripting (Version_Check.script)

A denial of service vulnerability has been found in the way the multiple overlapping ranges are and led by the Apache HTTPD server. An attack tool is circulating in the wild. Active use of these tools has been observed [12].

Impact: The attack can be done remotely and with a modest number of requests can cause very significant memory and CPU usage on the server.

Recommendations: Remote Denial of Service.

Backup Files

Severity	**Medium**
Type	Validation
Reported by module	Scripting (Backup_File.script)

Possible backup file was found on server which is generally created by developers to back up their work.

Impact: Backup files can contain script sources, configuration files or other sensitive information that may help a malicious user to prepare more advanced attacks.

Recommendations: Remove the file(s) if they are not required on your website. As an additional step, it is recommended to implement a security policy within your organization to disallow creation of backup files in directories accessible from the web.

Admin Page of the website can be searched using strings. Some of the examples are [11].

- adm/
- admin/
- admin/account.php
- admin/account.html
- admin/index.php
- admin/index.html
- admin/login.php
- admin/login.html
- admin/home.php
- admin/controlpanel.html
- admin/controlpanel.php
- admin.php
- admin.html
- admin/cp.php
- admin/cp.html
- cp.php
- cp.html
- administrator/
- administrator/index.html
- administrator/index.php
- administrator/login.html
- administrator/login.php
- administrator/account.html
- administrator/account.php
- administrator.php
- administrator.html
- login.php
- admin/index.asp
- admin/login.asp

Admin page can be penetrated to guess the username and passwords by checking for the database specific default passwords implementing an ascii per mutative brute force password scanner. You could probably scan for weak passwords <= 5 characters in a single day. Use the Metasploit framework to check for public vunerabilities. Purchase a Zero-Day licence at one of the security research groups. I would recommend Vupen.

Step 3: Exploitation and Gaining Access

All these results have shown that website is vulnerable to attacks and further these vulnerabilities could be exploited using Metasploit Pro and Linux Backtrack.

As this test was conducted purely for the penetration testing research work and brute force, intensive penetration tests require client permissions therefore intensive tests were not conducted but it is the proof that still there are thousands of websites presents on the internet which are running successfully but not secured at all.

Conclusion

Past two decades have shown that system compromises are on the rise, so we must guard against them using the methods available to us. We are using firewalls, intrusion detection tools, better security policies, and all the other defences quoted in this paper but we should also use Penetration testing as one of the arsenal available for ensuring superior secure systems. NASSCOM statistics show that Penetration testing market is at 8% of total Software Testing market and growing since companies have realized the importance of Penetration Testing. It is important to acknowledge that a good follow-up plan to correct any vulnerability found is just as important as detecting them and one of the Best ways to detect and avoid them is Penetration Testing.

REFERENCES

[1]http://en.wikipedia.org/wiki/Data
[2] The Economic Impact of Cyber-AttacksApril 1, 2004Brian Cashell, William D. Jackson, Mark Jickling, and Baird WebelGovernment and Finance Division
[3]http://www.securityfocus.com/news/2445
[4]http://www.informationweek.com/news/galleries/security/attacks
[5]https://www.owasp.org
[6]Twenty Critical Controls for Effective Cyber Defense:Consensus Audit GuidelinesVersion 2.1: August 10, 2009
[7]CRS Report for Congress, USA
[8]CERT,2003, CERT/CC Statistics 1988-2002, 2003, April

15,[http://www.cert.org/stats/cert_stats.html#incidents. CERT, 2003, CERT/CC Statistics,2003, [http://www.cert.org/stats/cert_stats.html.
[9]Kevin Poulsen, April 12 2001, War Driving by the Bay, Securityfocus.com
[10]Anne Saita,May 2003, Antiforensics: The Looming ArmsRace, Information Security, Vol. 6, No. 5, p.13.
[11]www.breakthesecurity.com
[12]http://seclists.org/fulldisclosure/2011/Aug/175

Other References (Books)

1. The Basics of hacking and penetration Testing, Patrick Engebretson

2. Gray Hat Hacking The Ethical Hacker's Handbook, Third Edition, Allen Harper, Shon Harris, Jonathan Ness, Chris Eagle, Gideon Lenkey, and Terron Williams

About the Author: Harish Chaudhary

Harish Chaudhary is a young, motivated and dynamic Software Quality Engineer at QA InfoTech with just about one year of experience. He's passionate for Security and Penetration Testing to safeguard data from hackers. He's a keen orator and has won himself a position by participating in testing conferences organized internally at QA InfoTech. He is a member of Codeproject.com since 2008 and is working as a freelance Technical writer. Few of his articles have also made it to a Software Testing Magazine – 'Testing Circus'. For more information on this paper, please write to us at info@qainfotech.com