International Conference on Information & Communication Technologies: From Theory To Applications, 19-23 Apr 2004, Damascus, Syria.

Development of Penetration Testing Model for Increasing Network Security

Rahmat Budiarto, Sureswaran Ramadass, Azman Samsudin, Salah Noor
Network Research Group School of Computer Sciences, USM11800 USM, Pulau Pinang, Malaysia
E-mail: {rahmat, sures, azman}@cs.usm.my, salah@nrg.cs.usm.my

Abstract

With the emergence of network globalization and advent of Internet being the major tool for international information exchange and platform for the future, security has always been the most talked about topics. Much emphasis has been given to security, due to the fact that networks are very much vulnerable to Denial of Service attacks or security and access compromise. Network administrators have often tried their best by improving their network security, however with rapid surface of new exploits; the best way of ensuring that the system is secure is to attempt penetration testing. This would be the most effective way to find exploits and to proof whether a system is vulnerable. Penetration testing often allows the security analyst to find new vulnerabilities.

1. Introduction

The goal of penetration testing is to identify the exploits and vulnerabilities that exist within an organisation's IT infrastructure and to help confirm the effectiveness – or ineffectiveness – of the security measures that have been implemented. Indeed, there is rarely a better way to justify additional funding for security controls than by physically demonstrating the flaws that exist in operational systems. A board of directors will instantly appreciate the value of security once they've witnessed the exposure of confidential information by a successful penetration test.

It's important, though, that penetration testing should model real world attacks as closely as possible. In practice, whilst a real world attacker would typically spend many months researching a target, a penetration tester will rarely be afforded this luxury. They need to complete, in several days, the activities that a real attacker would spend considerably longer conducting.

This is why it is both useful and good practice, to examine the internal configuration of systems before attempting an external penetration test. It enables the tester to quickly gain an insight into an organisation's IT infrastructure and to model the months of research an attacker would expend on creating a knowledge base about a potential target.

Threats come from many different sources. In practice, around 70% are either internal incidents or are accidental or malicious in nature. The remaining 30% are the result of externally-related incidents. The most serious security breaches are, more often than not, carried out by insiders who have taken advantage of their intimate knowledge of a company's systems [3].
Penetration testing should, therefore, model the attack profiles of potential threat sources in order to accurately determine the possibility of an attack succeeding. These are often split down into several, clearly defined categories and are listed below in Fig.1. Each of the individuals described have different attack profiles and these have to be carefully modeled in order to re-create attack scenarios that are as realistic as possible.

2. Penetration Testing Model & Method

There are two distinct models for penetration testing - the Zero Knowledge test and a Full Knowledge test [10]. With the former, the tester is given no insight into the target systems under investigation. With a Full Knowledge test, however, the tester is given complete information about them.

Zero Knowledge tests are useful when trying to ascertain how vulnerable systems are from the attack profile of the Script Kiddie. These are the most common type of attackers and are generally regarded as no more than Internet vandals. They typically attack the easiest targets they can find and with complete disregard. They rarely conduct any research and normally start an attack as soon as the target is acquired.

A Full Knowledge attack sets out to accurately model the attack characteristics of a Master Hacker or Malicious Insider [8,9,11]. This is because both of these individuals will already know a great deal about an organisation's systems (Malicious Insider) or will carry out extensive research (Master Hacker) in order to identify the best ways of attacking a system.

All penetration tests use a similar methodology regardless of the actual attack profile that is being simulated. Target acquisition is the process by which the tester gains as much information about a target as possible. This can be done in several ways such as scanning a web site for names, photographs or contact telephone numbers.
The use of online whois databases can often retrieve vast amounts of information such as system administrator IDs and network addresses. Services such as Companies House can also offer pertinent information regarding

management employees. Once the network location has been identified, the tester can then utilise port and network scanners to identify available services and the topology of a network. Tools such as nmap, fping and icmpquery provide a plethora of information which can additionally be used to develop a plan of the network. Nmap provuides the functionality for TCP fingerprinting and can be used to help identify the operating system running on a particular network server. It can often even reveal the release version that is currently use.

Enumeration is the process of attempting to obtain user names, network share and application version information from the services running on a server (eg. Apache 1.3.X, BIND 8.2.1). It's achieved through the interrogation of network systems and banner grabbing or may involve the use of tools such as *gnit* and *netcat* in order to more intrusively retrieve system information from hosts.

Script Kiddie
Has limited or no knowledge of how computer systems work. They rely on pre-written exploits and vulnerability scanners to find and realise vulnerabilities release version that is currently in use.
Master Cracker
Has intimate knowledge of IT technology and system code. They find original vulnerabilities, write customized exploits and spend much of their time learning and finding flaws in new technology.
Malicious Insider
Does not necessarily know much about IT systems but does know a lot about YOUR system. This enables them to attack a system at its most vulnerable point.
Once the tester has built up a satisfactory
Naive employee
Generally damages IT systems through an inability to correctly operate even the simplest applications.

Figure1: Profiles of potential attackers

3. Exposing Vulnerability

Once the tester has built up a satisfactory library of system information, the vulnerabilities have to be found. This can be performed by manually matching the applications present to publicly available vulnerability lists such as CERT (Computer Emergency Response Team) and Bugtraq [6,7].
Using this manual method results in a long and drawn out procedure. Automated tools such as ISS's vulnerability scanner are available, however, and these can rapidly provide a comprehensive list of the vulnerabilities that exist on a target system.
Whilst useful, tools such as these are unlikely to identify the most critical vulnerabilities that affect a specific system. After all, every IT system is unique and vulnerability scanners rarely take this into account. What this means in practice, is that the value of automated tools is superceded by the experience and guile of an experienced penetration tester.

4. Stepping Outside of the Box

Automated tools are designed to operate under the same set of rules as the test target. It is only by stepping outside of this environment that it becomes possible to find 'holes' in a system. The penetration tester must, therefore, be prepared to step outside of the problem rather than merely operating within it. This, in fact, is one of the main differences between Script Kiddies and penetration testers. Script Kiddies will often pass over this aspect of testing by adopting a 'shotgun' approach compared to the 'scalpel' technique of the penetration tester. Script Kiddies will utilise every resource at their disposal without any concern as to whether they work. As long as one of the scripts succeeds, they rarely care about the rest. The penetration tester, however, has a full list of potential vulnerabilities and system information and this can be used to select exploits that will be run against the target system. The user names and passwords –collected at the enumeration stage – now become useful as they can be employed to gain access to the target.

5. Offensive Operations Model

To properly assess the security of a system, understandings of the different phases of a successful attack or intrusion are necessary. By understanding the risk of exploitation, both can be applied to a structured list of possible controls to assess the current state of security, and the directions that need to be investigated.

All successful intrusions share the following characteristic phases [2]:

1. Reconnaissance
2. Assessment and Strategy
3. Exploitation / Invasion
4. Maintaining Access
5. Operations

Hackers place different priorities on each stage. In essence, the more time spent on one step ensures better results in the following steps. Reconnaissance can go undetected for considerable lengths of time and the Assessment and Strategy stage is often completely undetectable, as it is usually done without contact with the target. Each phase is conducted in such a way as to ease the way for the next step, and lower the chance of getting caught.

In each of those stages, there is the risk of exploitation. The types of exploits are:

1. Confidentiality (can privacy be compromised?)
2. Integrity (can data accuracy be compromised?)
3. Availability (can data accessibility be compromised?)

The stage of the attack plus the type of exploit identifies the risk. As an example, in Reconnaissance a hacker is primarily collecting data. There is no intention to alter data integrity or availability, although

Confidentiality is affected. Therefore at the Reconnaissance stage of the attack, there is a risk of Loss of Confidentiality. At the other end of the scale, the Operations stage is when the hacker performs his or her intent. If he is spying, Confidentiality is at risk. If he is malicious and intending on causing damage to the company, Integrity and Availability are at risk. This may also be the case if the hacker intends only to spy, but mistakes made along the way have affected Integrity and Availability.

Strictly speaking, a control is a mechanism to reduce risk. This may entail blocking data flow to outside networks, ensuring data integrity, or maintaining its accessibility. Controls also provide functions to notify when an attempt has been made to circumvent allowable access, and an audit trail to accurately document differences. Most controls are focused on a limited number of threats or vulnerabilities, and singularly can be defeated. Because of this, a robust suite of controls is necessary to mitigate risk.

There are 6 categories of controls [2]:

1. Prevention (will the controls sufficiently prevent an incident?)
2. Detection (will the controls sufficiently detect an incident?)
3. Containment (will the controls sufficiently contain an incident to a limited target?)
4. Eradication (will the controls sufficiently allow the vulnerability to be fixed in a timely and accurate manner?)
5. Recovery (will the controls sufficiently allow recovery without re-introducing vulnerability?)
6. Follow-up (is there a process to document an incident? post-mortem reporting?)

Given the 3 types of exploits and the 5 characteristic phases of a successful intrusion, policies, procedures and assessments can be developed around the 6 categories of control. For each stage of an attack, at least one of the risks of exploitation are used, and often in several possible ways. In other words, for each stage of the attack, each of the 6 categories of controls are tested and weighed against their potential associated risks.

5.1. Questioning the Controls around a Given Risk

In the following series, each stage of an attack is listed, along with each control category. The sample questions are provided to demonstrate assessment of the associated control. Throughout the actual procedure, questions are similar in spirit, and tailored to appropriately assess the control being tested.

5.1.1. Reconnaissance, Assessment and Strategy

Reconnaissance, or Recon, is the act of scoping out a target [3]. This information gathering stage is the most important step a hacker takes, and all key information is considered. The Assessment and Strategy stage is the sorting of the gathered data to piece together an idea of what the hacker is attacking. These two stages are assessed together because Recon is the part of the act that involves interaction of some sort with the target, and the Assessment and Strategy stage is usually done remotely by reviewing the gathered data.

When assessing controls around the Reconnaissance and Assessment and Strategy stages of an attack, each of the 6 areas of control need to be identified. Example questions are given, although in real-case scenarios, questions will be more directed to what is being assessed.

1. Prevention - Do server banners provide too much information about the system or network? Do login scripts behave identically for failed usernames as with correct ones? Is directory browsing through http disabled?
2. Detection - Is logging effective?
3. Containment - Could information obtained from the server provide information about neighboring systems and networks?
4. Eradication - Is a method in place to remove, block, or change data that may be used to create more specific attacks against the server?
5. Recovery - Can system functions "recover" old data that was previously removed to prevent reconnaissance?
6. Follow-up - Are post-mortem reports generated when there is potential evidence of data-mining? Is a record kept of all dates, times, IP addresses and suspected network-mapping activity for reference if a penetration or other crime is committed?

5.1.2. Exploitation and Invasion

Once a hacker has gathered enough information and has pieced together a reasonable amount of information about the network or system they are attacking, and have devised an initial plan of attack, it is then possible to begin the Exploitation and Invasion stage. At this point, the hacker uses the gathered knowledge and attempts to access the server through the channels that were found open .

When assessing the controls around Exploitation and Invasion, the following types of questions can be asked:

1. Prevention - Have the appropriate security patches been applied? How are buffer-overflows prevented? Is the access control sufficiently strong? Could the system play host to a Distributed Denial of Service attack?
2. Detection - Would a successful penetration be detected? Is a method in place to notify individuals who are in authority to react to an incident?
3. Containment - Can penetration of a single service cause compromise in other servers? Can a single service be used to control the entire system?

4. Eradication - Do controls allow for scalability when a compromise dictates change? Do controls protect potential forensic data?
5. Recovery - Can the system be brought back online in a short period of time following a worst-case intrusion, without re-introducing the vulnerability?
6. Follow-up - Are detailed reports of an incident and its mitigation generated by the system?

5.1.3. Maintaining Access

Once a hacker has penetrated the network (or if the hack is an inside job) steps are usually taken to make future accesses easier to conduct. This often includes installing a back-door program, but sometimes may be something as simple as setting up a home base under a seldom-used account name or identifying a mis-configured user account with suitable permissions to use to regain entry.

When assessing controls that limit hackers from increasing their level of access to improve re-entry, the following types of questions can be asked:

1. Prevention - Is effective change detection software installed and enabled? Is the system administrator alerted when access levels and permissions have changed?
2. Detection - What features are running on the system to detect back-door programs, or critical file-system changes?
3. Containment - Could a backdoor installed on the system be leveraged to attack another system?
4. Eradication - Are features in place for reassigning access levels or permissions if they have been changed? Are back doors such as Back Orifice automatically removed?
5. Recovery - Is sufficient change control in place to void accidentally reintroducing a backed-up version of a back door mechanism? Do these features inhibit other system functions negatively?
6. Follow-up - Are detailed reports of suspicious fie permissions generated? Are logs sufficiently detailed to investigate the source of back doors or Trojan files?

5.1.4. Operations

This is the most dangerous part of a penetration - the hacker has all the access required to carry out their agenda. If it is a spy operation, data could be sent to a remote collection repository. If it is a system-mapping reconnaissance mission, existing levels of access may be used to compromise more systems on the network.

When assessing controls that limit illicit operations on systems and the network, the following types of questions can be asked:

1. Prevention - Is new executable code added to the system disabled and quarantined?
2. Detection - Is an alarm mechanism in place when operations are detected? Is someone with enough authority to investigate and react appropriately notified of a potential operation?
3. Containment - Is the system architecture robust enough to limit a hacker to a single environment? Does manipulation of a single target affect unrelated functions? Is a mechanism in place to minimize and control damage?
4. Eradication - When operations are detected, is a mechanism in place to end the activity and deal with it appropriately?
5. Recovery - Can a system be coerced into accidentally recovering a back-door mechanism?
6. Follow-up - Is a mechanism in place to follow an incident from the moment of suspicion to the point that the case is considered closed?

6. Conclusions

No matter what the threat, a professional penetration test should accurately model the attack characteristics of the profiles discussed. A methodical and scientific approach should be used to successfully document a test and create reports that are aimed at different levels of management within an organisation.

Penetration testing should never be regarded as a one-off service. Systems change, threats emerge and business strategies evolve. Testing should be repeated at frequent intervals and particularly following major changes to an IT infrastructure. It's also important to remember that penetration testing is but just one form of testing and any organisation should develop an overall security testing strategy that is tailored to the threat models and security policies of their organisation.

As can be seen, using the Offensive Operations Methodology can uncover security flaws in any feature, configuration or trust relationships, whether they be technical, architectural, design or policy related. A gap in any one of the risk-control areas during any phase of an attack, as described above, is potential security vulnerability. All the attacks and test procedures are based on the existence or apparent existence of these gaps.

7. References

[1] Paul Midian, "An Insight White Paper Penetration Testing", Insight Consulting, URL: www.insight.co.uk, January 30, 2004

[2] Karsten Johansson, "Offensive Operations Model", KSAJ Inc, Version: 1.0 Public Release, Posted: http://www.penetrationtest.com, August 7, 2001.

[3] Abreu, Elinor. "No Budget? That's No Excuse for Not Testing Your Network Security". April 25, 2001. The Industry Standard. URL: http://www.thestandard.com/article/0,1902,23991,00.html (17 December, 2003)

[4] Bar-Gad, Izhar. "Identifying the 10 Most Common Application-Level Hacker Attacks".September 17, 2001. Yahoo! India Technology. URL: http://in.tech.yahoo.com/010917/22/14vxo.html December 17, 2000

[5] Christensen, Paul. "An Introduction to Nessus", May 7, 2001. LinuxSecurity.com, URL: http://www.linuxsecurity.com/feature_stories/feature_story-86.html December 17, 2003

[6] De Beaupre, Adrien. "Know yourself: Vulnerability Assessments". SANS Institute Information Security Reading Room. URL: http://www.sans.org/infosecFAQ/audit/know.htm December 17, 2003)

[7] Fyodor. "The Art of Port Scanning". September 6, 1997. Insecure.org URL: http://www.insecure.org/nmap/nmap_doc.html (17, December 2003)

[8] Herman, Ben. "Routine External and Internal 'Hacking', An Important Part of Information Assurance". April 19, 2001. SANS Institute Information Security Reading Room. URL: http://www.sans.org/infosecFAQ/attack/routine.htm (17 December, 2003)

[9] Koerner, Brendan I. "Showdown at Hacker Gulch". June, 2001. Business 2.0 URL: http://www.business2.com/articles/mag/0,1640,14779,FF.html (17 December, 2003)

[10] Kurtz, George & Chris Prosise. "Penetration Testing Exposed". September, 2000. Information Security Magazine. URL: http://www.infosecuritymag.com/articles/september00/features3.shtml (17 December, 2003)

[11] McClure, Stuart. "Digital Battlefield". 2001. Foundstone URL: http://www.foundstone.com/cgi-bin/display.cgi/?Content_ID=180 (17 December, 2003)