SANS

www.sans.org

MANAGEMENT 414

SANS TRA...

PROGRAM ... SSP

CERTIFICATI...

Domain 6

Security Assessment and Testing

(Designing, Performing, and Analyzing Security Testing)

The right security training for your staff, at the right time, in the right location.

MANAGEMENT 414
SANS Training Program for the CISSP® Certification Exam

Domain 6:
Security Assessment and Testing

MGT414: SANS Training Program for CISSP® Certification

This page intentionally left blank.

Domain 6: Security Assessment and Testing

- **Security assessment strategies**
- Technical security testing
- Security audits and key security processes

This page intentionally left blank.

Security Assessment

- For practical and regulatory purposes, organizations must assess their security posture
- To gain a holistic view of security requires
  - Technical security testing
  - Security process assessment
  - Security audits
- Analyzes the entire network from the inside and tries to find the weaknesses
- Offers a complete list of risks against critical assets

Security Assessment

A security assessment is a more complete view of a company's network security. It analyzes the entire network from the inside and tries to find the weaknesses and gives a complete list of risks against critical assets. This type of test is usually recommended if you want to come up with a roadmap and better understand your security risks.

At the conclusion of a security assessment, you get a prioritized list of what the critical risks to your assets are, what the likely hood of the risks occurring are, what the costs are, and what the cost to fix the risks are. Based on this information, management can make the proper decision about what level of risk they are willing to accept. Therefore, a security assessment helps you manage risk in a more "holistic" manner.

Security Testing

- Technical security testing involves overtly looking for potential security weaknesses
- The most basic and common type of security testing will be the vulnerability assessment
- Vulnerability assessments are not the only approach to technical security testing
- Other approaches include
  - Network penetration testing
  - Web application penetration testing
  - Source code analysis

Security Testing

Technical security testing is an extremely common way to assess security posture. Most organizations are well versed in the use of vulnerability scanning tools to review the risk associate with missing patches.

Though vulnerability assessment is the most commonly employed type of test, there are others that will also be presented within this section.

Vulnerability Assessment

- Scanning key systems looking for a set list of vulnerabilities
- Usually done to look for common or known vulnerabilities
- Performed using a vulnerability scanner tool
- This assessment can be performed in-house, by a third-party, or often both

Vulnerability Assessment

A vulnerability assessment or VA occurs when you scan key servers to look for a set list of vulnerabilities. It is usually done to look for common or known vulnerabilities and it is usually performed using a vulnerability scanner tool.

There is no point in configuring the scanner to hit all of your addresses unless you are in a small organization - scan one subnet at a time, one workgroup at a time, or whatever makes sense. This way, you won't have an overwhelming number of vulnerabilities to fix.

If you do scan the whole facility, you will get a huge list of problems. With a huge number of problems, people in the organization will talk about fixing them, but because there are so many problems, they likely will not past the promise stage. This is very dangerous. After you run the scan on a large scale, you get a huge printout of all the problems and some of them are flagged as "very" serious, some are just somewhat serious, and so on. You present the list to management, tell them it is the end of life as they know it if the problems aren't fixed. Management agrees, they task people, there are meetings, everyone agrees to get things fixed, and then they run into deadlines and emergencies. The problems never get fixed. Now you can't play that card again. After all, the organization is still in business! If you run another scan, no one will take it seriously.

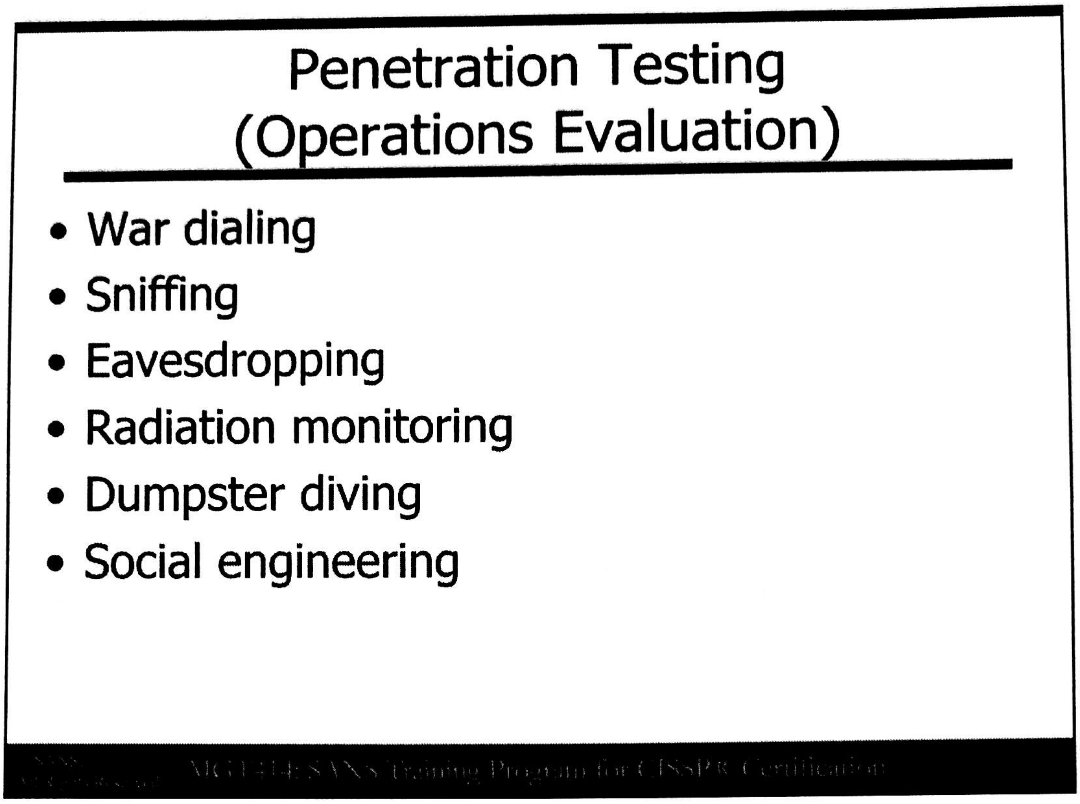
Penetration Tests

- Penetration testing picks up where vulnerability assessment stops
  - Rather than mere identification of flaws, penetration tests seek to exploit the vulnerabilities
- Simulates an attacker trying to break into a network
- Determines whether a site is susceptible to attack
- Are only as good as the person/tool behind the test

Penetration Tests

A penetration test simulates an attacker and tries to break into a network. The goal is to determine whether a site is susceptible to attack. However, this test is only as good as the person who performs the test. It does not give a complete view of the security of a network.

Let's discuss some of the ways a penetration test can be done. A penetration test is sometimes completed at the conclusion of a vulnerability assessment and is used to determine the validity of any identified vulnerabilities. This ensures that all false positives are eliminated if the vulnerabilities are exploited. Penetration tests are sometimes run in lieu of a Vulnerability Assessment and are conducted entirely from outside the network being tested, from the perspective of a true hacker. They can evaluate the effectiveness of your security perimeter, including routers, firewalls, servers, and any other perimeter security devices.



Penetration Testing (Operations Evaluation)

After the operational security plan is in place, it must be tested. *Penetration testing* is the process of examining the limitations of the security measures in place. Some tests include:

- *War dialing* - attempts to attack the systems via dialing all the phone numbers in an exchange.
- *Sniffing* - passively monitors network traffic for network knowledge, such as passwords

Beyond Port Scanning

- Port scanners perform more than simple TCP/UDP port scanning
  - Initial focus was on pure port scanning
- Newer features include
  - Service identification
  - OS identification
  - Tactical vulnerability assessment
- Scanning engine a vehicle to rapidly assess particular configuration weakness or flaws

Beyond Port Scanning

Port scanners have matured beyond merely sending stimulus to TCP and UDP ports to determine if the port is open or not. Operating System and service fingerprinting are common capabilities. Beyond fingerprinting systems and services, now port scanners can even be employed to check for simple configuration weakness or vulnerabilities.

Without question, nmap is the king of port scanners. It initially focused on port scanning only, but has added a lot of additional functionality, including service and operating system identification, and the Nmap Scripting Engine (NSE).

NSE takes Nmap from a simple port scanner, to a network vulnerability scanner. NSE scripts are written in Lua, an embedded scripting language.

More information on NSE is available at: http://nmap.org/book/nse.html (http://cyber.gd/414_62).

OS or Service Fingerprinting

- Fingerprinting seeks to identify the version of the operating system or services running on a target system
- Active fingerprinting sends packets to determine OS and service versions
  - For example: **`nmap -A 10.20.30.2`**
- Passive fingerprinting is read-only
  - Uses TTLs, IPIDs, sequence numbers, and even Layer 7 packet data to determine system details
  - For example: **`p0f -s capture.pcap`**

OS or Service Fingerprinting

Fingerprinting goes beyond simple port scanning, seeking to identify the operating system, application, and service levels. Active fingerprinting sends network traffic.

There are a number of ways to determine OS and service version levels, including banners, packet characteristics, web page content, etc. Nmap is one of the best active fingerprinting solutions, see http://nmap.org

Passive fingerprinting is read-only, inspecting packets passively on the wire, or via a pcap file.

p0f (by Michal Zalewski) is one of the best passive fingerprinting solutions. It was recently updated after a long hiatus; the newest release at the time of printing is p0f v3. See: http://lcamtuf.coredump.cx/p0f.shtml (http://cyber.gd/414_63)

Vulnerability Assessment

- Vulnerability assessment determines weaknesses in a system
  - This type of test does not include exploitation
  - Focuses exclusively on the "vulnerability" portion of the risk = threat x vulnerability equation
- Vulnerability assessment uses a vulnerability scanning tool, such as
  - Nessus (Tenable)
  - OpenVAS (Open Source)
  - Nexpose (Rapid7)
  - Retina (BeyondTrust)

Vulnerability Assessment

More information about the vulnerability scanning software listed above can be found at:

- Nessus by Tenable
- OpenVAS
- Nexpose by Rapid7
- eEye's Retina

Vulnerability Scanners

- Vulnerability scanners go beyond port scanning
  - Determine available applications and services
  - Determine their versions
  - Determine if they are vulnerable to exploitation
- Methods to determine vulnerabilities:
  - Determine version information, look up vulnerability in database
  - Interrogate the system, modeling insecure behavior

Vulnerability Scanners

Vulnerability scanners go beyond simple port scanning, and attempt to determine vulnerable services and applications.

There are a variety of methods for determining whether a service or application is vulnerable. A simple (and lower risk) method is to determine the service or application version (often by inspecting a banner), and looking up the version information in a vulnerability database.

A more invasive approach is interrogating the system, sending it traffic, and attempting to model insecure behavior. This is riskier, since the chance of crashing an application or service is higher.

A higher-risk method is actually exploiting the service, which is what penetration testers will do. This is riskier because there is a possibility the service, application, or system will crash. Data may also become corrupted, a violation of integrity. This would be considered exploitation, a step beyond vulnerability assessment.

Attacking Tools and Frameworks

- Once vulnerable services have been discovered, attacking tools and frameworks may be used to exploit them
- Examples include
  - The Metasploit Framework (metasploit.com)
  - Core Impact (coresecurity.com)
  - Immunity Canvas (immunitysec.com)
- Dedicated Linux-based distributions include Backtrack and Kali

Attacking Tools and Frameworks

The next step beyond vulnerability assessment is exploitation. Core Impact, Immunity Canvas, and Metasploit are three popular options. These tools bundle functionality from network enumeration, to vulnerability assessment, to exploitation: all under one hood.

The old Metasploit slogan "Point. Click. Root." illustrates the power of these tools. Core Impact, as one example, is so powerful that an amateur can generate impressive results. These are truly powerful tools in the hands of an expert.

Penetration Testing

- Penetration testing is the process of hiring a whitehat to penetrate an application, system, or network
  - In a sense, this simulates a malicious adversary with the same goal
- A penetration test is narrower than an overall security assessment

Penetration Testing

A penetration test is a narrow, but useful test when conducted by skilled professionals. If a whitehat can compromise an application, system or network then a blackhat could likely do the same.

The potential scope of a penetration test can be quite wide, and include any of the following and more:

- Client-side exploitation
- Server-side exploitation
- Web application exploitation
- Wireless exploitation
- War dialing
- Attempted physical access
- Social engineering

A successful penetration test is a useful business tool, and can be used to effect positive change. But what happens if the penetration tester fails to compromise a system? Does that mean the system is perfectly secure, or did the tester lack the skill or time to compromise it? Or was the scope overly narrow?

Overall Penetration Testing Process

- Business processes
  - Scope, Rules of Engagement, etc.
- Reconnaissance
- Scanning
  - Vulnerability assessment
- Exploitation
- Post exploitation

Overall Penetration Testing Process

The penetration testing process begins as a business process. What is in scope: client-side attacks, server-side attacks, social engineering, etc.? When will the penetration test take place? Explicit legal permission must be granted and this is often a challenging process). Contracts are required, and lawyers for both the penetration tester and the client organization must be part of that process.

Once the business is taken care of, the process follows the steps we have been discussing in this section: reconnaissance, scanning, vulnerability assessment and exploitation. Additional post exploitation steps, such as password cracking, may also be used, as we will discuss next.

Additional Security Testing Methods

- Additional methods and tools for testing information security include:
  - Black box and white box testing
  - Code review
  - Fuzzing
  - Web application testing
  - Interception proxies
  - Phishing campaigns
  - Password assessment
- May be used independently, or as part of other tests, such as penetration tests

Additional Security Testing Methods

We will next discuss additional methods for testing the security of an application, system or network.

These methods are sometimes used independently, or as part of larger tests (such as penetration tests). For example: after receiving explicit permission, you may assess the strength of your organization's passwords by cracking them.

A penetration tester could also use password cracking as part of a penetration test, such as post-exploitation activity after compromising one host.

Black Box vs. White Box Testing

- White box testing is a software testing method that uses internal algorithms and information to conduct the test
  - Source code review
- Black box testing begins with no inside knowledge of an application
  - Can be used against compiled code with no access to source
  - Fuzzing is usually a black box process

Black Box vs. White Box Testing

In addition to software testing, these terms also have specific meaning within penetration testing:

Black box penetration testing begins with no inside knowledge of an organization and targets. Also called a 'zero knowledge' test.

White box pen testing uses inside information at the outset. Also called a 'full knowledge' test.

Gray box is a combination of black and white box penetration testing. Also called a 'partial knowledge' test.

Source Code Review

- Code review is a white box testing approach that attempts to discover security vulnerabilities by inspecting the source code of a target application
- For example: certain C functions are commonly associated with buffer overflows
  - `gets()` , `strcpy()`, `strcat()`, etc.
- Insecure use of such functions can identify vulnerabilities
- Though compilers increasingly include basic security checks, more thorough analysis is required

Source Code Review

A formal code review requires the inspection of every line of code, and requires the reviewer to fully understand what the code is doing. This is called "heavy" code review.

Newer code review methods are designed to be "lightweight," and are less intensive than formal code review.

Pair programming is a lightweight programming, where programmers sit side-by-side. One programs while the other reviews the code, and the two can switch roles. XP (eXtreme Programming) uses paired programmers.

Another lightweight method is called tool assisted, where code review software searches for insecure code examples. Code review software solutions include: CodeCollaborator, Codestriker, Crucible, Jupiter and Review Board

Fuzzing

- Fuzzing is a black box process that sends unexpected output to computer programs
- For example: A program says:
  - `Enter your username:________`

- A fuzzer could send:
  - A
  - AA
  - AAA
  - ...
  - AAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAAAAAAAAAAAAAAAAA
    AAAAAAAA
  - Etc...

Fuzzing

Cramming is the process of sending unexpected input into applications. For example: if a program asks for username, a user could attempt to type 1,000 characters, to see if the program crashes. The Python programming language is commonly used for writing fuzzing scripts.

Fuzzing (also called fuzz testing) is automated cramming. Many fuzzers send repeating (and lengthening) strings of characters, such as one "A", then 4, 8, 16 ... 1024, etc.

For example: a program handles 1,020 characters cleanly, but crashes after 1024. This is an indication that the programmer did not protect bounds, and there are a roughly 1,020 characters between the start of the buffer and the return pointer. These are the first steps to developing an exploit to smash the stack, place machine code on it, overwrite the return pointer, and execute the code.

Web Application Testing

- Code analysis and fuzzing are commonly employed in testing web applications as well
- HTTP interception proxy intercepts web data in real time
  - The primary tool in performing dynamic web application penetration testing
  - Can view/edit cookies, hidden form elements, authentication data, etc.
- Beyond the proxy, the dynamic web application scanner attempts to automate assessing the security of custom web applications

Web Application Testing

Assessing custom web applications is vital given how common they are employed for performing critical business functions. They also provide the front-end by which sensitive data is often accessed.

An HTTP interception proxy is like "bullet time" in the movie *The Matrix*: they allow a researcher to pause events such as the transmission of a session cookie, alter it, and then continue transmission.

The tester configures his/her browser to proxy via the interceptor, and then the interceptor connects to the target website.

Note that this process works with both HTTP and HTTPS. The tester uses SSL/TLS to connect to the proxy, and the proxy will then use SSL/TLS to connect to the target web site. The tester's browser will likely display a certificate warning (the browser will see the proxy certificate, and not the target web site). Since the researcher is essentially conducting a man-in-the-middle attack against the browser, this warning may be ignored.

Dynamic web application scanners are another, more automated, method for assessing web applications' risk.

Password Assessments

- Password guessing is attempting to authenticate as a user by guessing their password
- Password cracking involves performing an offline attack vs. password hashes
- A hash algorithm is not reversible
- Password crackers run a hash algorithm forward many times
  - Takes an input, such as a list of dictionary words
  - Runs each input through a hashing algorithm
  - Compares output to stored hash

Password Assessments

Password cracking is an offline process which attempts to match a password with its hash output.

Password guessing is a related, but different process. Password guessing is an online process, where the attacker repeatedly attempts to authenticate to a network service, such as sshd.

Password cracking requires password hashes. These are commonly acquired from the file system (such as the /etc/shadow file on Unix/Linux), or from the network (via sniffing, or by tracking users into authenticating to a system designed to collect the hashes).

Some password hashing algorithms, such as Microsoft LANMAN, do not use salts. This means the password "camel" always hashes the same way. A salt is a random string that is hashed along with the password, which is the method modern Unix/Linux systems use. This means the password "camel" will hash differently for multiple users. This also makes the use of pre-computed password/hash pairs (such as a rainbow table attack) much less practical. We will discuss rainbow tables next.

Types of Password Cracking

- A dictionary attack hashes the words in a dictionary
  - cat, dog, camel, etc.
- An incremental attack starts with a dictionary, and then adds characters
  - camel1, camel2 ... camel99, etc.
- A brute force attack hashes every possible password
  - aaaa, aaab, aaac ... zzzz
- A rainbow table contains pre-computed password/hash pairs

Types of Password Cracking

A dictionary attack typically uses an actual dictionary as the input. A smart attacker uses a dictionary tuned to the target audience, adding a Spanish dictionary for targets with Spanish-speaking users, for example.

A large collection of dictionaries is available at: ftp://ftp.ox.ac.uk/pub/wordlists/

Incremental attacks play off the fact that many people will append a single character to a word, and use that as a password. In cases of forced password changes, many users will choose their old password, appended with 1. When forced to change that, 2 is appended instead, etc.

A brute force attack will always succeed, the only question is time. The hash algorithm should be strong enough to withstand years of attack, when a well-chosen password is used.

Rainbow tables use pre-computed tables of passwords and matching hashes Although they appear to act as a database lookup, they are quite different internally. Rainbow tables use a time/memory tradeoff to optimize storage. Philippe Oechslin's paper *Making a Faster Cryptanalytic Time-Memory Trade-Off* discusses this concept[1].

[1] http://lasecwww.epfl.ch/php_code/publications/search.php?ref=Oech03 (http://cyber.gd/414_64)

Client-Side and Social Engineering

- Client-side exploits typically depend on some degree of social engineering
  - End-users are the vehicle by which adversaries can introduce code to the vulnerable client-side application
- Assessing users susceptibility to social engineering can expose this often weak link
  - Also presents a valuable "teaching moment" to help make end users more aware of the threats

Client-Side and Social Engineering

Client-side exploitation being the predominate vector for initial exploitation challenges the traditional approach to security architecture and operations. One of the common elements of typical client-side exploitation campaigns involve the use of social engineering to induce an end user to open and attachment, click a link, or navigate to a predictable website.

Proactive attempts to assess end users susceptibility to social engineering is one of the most powerful security assessment techniques available.

Phishing Campaigns

- Phishing e-mails represent a major delivery vector for attacks
- Proactive phishing campaigns involve intentionally sending end users suspicious, yet harmless, e-mails
  - The e-mails should mirror current adversary tactics to be most effective
- Not a replacement for patching, but proactive internal phishing campaigns can be effective
  - At better assessing the likelihood of the success of this threat vector
  - At increasing awareness of end users to adversary techniques
- Primary goal is to increase security posture rather than shame or punish end users

Phishing Campaigns

Phishing e-mails have long been, and remain, a preferred method for client-side exploitation. Adversaries supply links or attachments in an e-mail that is crafted to induce the user to click the link or open the attachment.

A useful security testing approach involves intentionally sending end users suspicious e-mails that have been defanged. The goal is twofold, assess the organization's susceptibility and to provide an impactful opportunity to increase security awareness.

Security Reports

- Merely enumerating deficiencies does little to increase an organization's security posture
- The goal of security testing is to
  - Understand weaknesses
  - Prioritize and perform remediation
  - Seek to determine the underlying cause of the flaw
- Security testing can cause harm to an organization, so testing without remediation could easily achieve more harm than good
- Providing report data to appropriate management structure is also a key consideration

Security Reports

Many organizations are capable of producing far more security reports than they could every hope to consume, assess, and remediate. Care should be taken to ensure that the reports produced lead to actionable results in the form of increased security.

The end-goal of testing is not to produce a report, but rather to effect change. Scanning without remediation can easily do more harm to an organization than good for their security posture.

Domain 6: Security Assessment and Testing

- Security assessment strategies
- Technical security testing
- **Security audits and key security processes**

This page intentionally left blank.

Security Root Cause Analysis

- Without a doubt, enumerating flaws through security testing has its merits
  - Find a flaw -> fix a flaw
- An important question about the flaws discovered remains
  - What allowed for the flaw to exist in the first place?
- Fail to understand the underlying cause of a flaw, and chances are good the same flaw (or type of flaw) will crop up repeatedly
- Assessing security processes can help identify the root cause of the security failure
- Security audits, though not exclusively process-centric, also can discover security process issues

Security Root Cause Analysis

Security testing helps to identify flaws. These flaws primarily exist in already deployed systems and already developed applications. Finding and fixing security flaws is no doubt valuable, however, unless the underlying process failure is understood, most organizations are doomed to have repeat offenders.

Security Audits

- Audit implies that an organization is being measured against a defined standard
- Though not always, audits are very often associated with compliance efforts
- Auditors will determine compliance by assessing the organization against the defined standard
- Which standard depends upon many factors including industry and company size/disposition
  - Many organizations are expected to meet or exceed multiple standards

Security Audits

Security audits are a common occurrence, particularly in large organizations. Audits involve assessing an organization against a particular standard which they seek or are required to meet. Compliance oriented audits are a major consideration for many organizations.

Compliance itself is one of the most common goals of many security programs and CISOs. While compliance and security are not the same goal, there are many security standards that organizations are expected to achieve, depending upon both size and industry.

Third-party Audits

- Many of the audits security professionals deal with are Third-party audits
- Major regulatory compliance audits are typically performed by external entities
  - Even those that do not formally require 3^{rd} assessment are often performed by external assets
- Security personnel routinely assist third-party audits
  - Assisting in data collection to prove compliance
  - Addressing questions related to potential noncompliance
  - Working to help address audit findings

Third-party Audits

Most major audits, especially for regulatory compliance, will involve third-party auditors. This tends to be the case even when third-party auditors are not explicitly required.

Security professionals play several important roles in third-party audits. We frequently perform the following functions:

- Assisting in data collection to prove compliance
- Addressing questions related to potential noncompliance
- Working to help address audit findings

Key Security Processes

- The CISSP CIB specifically calls out a number of security processes that need to be assessed
  - Account management processes
  - Backup and recovery verification
  - Log review process
  - Security training and awareness
  - Disaster recovery and business continuity
- Audits also seek data on these processes

Key Security Processes

The CIB highlights a number of key security processes that need to be assessed/reviewed. Though often covered in more depth elsewhere, we will touch on the following:

- Account management issues
- Backup and recovery verification
- Security log review
- Security training and awareness
- Disaster recovery and business continuity

Account Management Processes

- Legitimate credentials and accounts are frequently used in security incidents
- Goal of this key process is to highlight particular aspects of account management to limit the likelihood
- Account revocation - accounts that are no longer needed must be disabled/removed
- Access granting - granting access should require appropriate document approval
- Privileged access - accounts with significant capabilities or access to sensitive data must be limited and monitored closely for abuse
- Access review/revocation - access to systems/data that are no longer needed must be revoked

Account Management Processes

Legitimate accounts are very often abused by adversaries during security incidents and data breach. Appropriate account management processes seek to limit the exposure.

Some examples of account management processes include:

- Account revocation - accounts that are no longer needed must be disabled/removed
- Access granting - granting access should require appropriate document approval
- Privileged access - accounts with significant capabilities or access to sensitive data must be limited and monitored closely for abuse
- Access review/revocation - access to systems/data that are no longer needed must be revoked

Backup/Recovery Verification

- Merely having a backup of data or systems is insufficient
- The backup is only useful if it can be recovered
- Unfortunately, many organizations discover backup failures when in the midst of need to recover
- A process for testing the efficacy of recovery is a key security process

Backup/Recovery Verification

Possession of a backup does not necessarily imply the ability to recover that backup. Many organizations realize this crucial point only when they attempt to actually perform recovery when needed.

To ensure an organization is not caught unaware with a failed backup right when they are in need, a key security process involves testing for successful backup and recovery of data and systems.

Log Review Process

- The majority of data breaches involve adversaries having access to an organization and remaining undetected for significant time
- In most cases, data indicating the compromise and activities was logged and available for the organization to discover
  - Though they typically fail to do so themselves
- A more thorough process of log review could help to identify adversary activity
  - Even perhaps before the successful exfiltration of data

Log Review Process

Reviewing security logs remains a valuable and underappreciated security process and control. The majority of major data breaches suggest that data about the adversary activity was available, though ignored, by the breached organizations.

Organizations routinely have weeks to months to detect and respond to an adversary in advance of successful data breach. However, overwhelmingly organizations fail to detect the compromise themselves, relying on 3rd parties to notify them of their own breach.

Security Training/Awareness Data

- Initial compromise of organizations overwhelmingly begins with exploitation of end users' systems
  - And typically involves the end user performing an action based upon inducement from the adversary
- Wielded effectively, security awareness can influence behavior to decrease the likelihood of users opening the attachment or clicking the link
- The key process data is ensuring that the organization is providing security awareness to all end users on a regular basis

Security Training/Awareness Data

Compromising end users via a social engineering attack is all too common. Security awareness attempts to help change user behavior to decrease the likelihood of clicking a link or opening an attachment.

Key process data on this point is ensuring that security awareness training is being performed for all end users on a routine basis.