Regelmäßige Penetrationstests beugen vor

Schwachstellen rechtzeitig erkennen

Viele Architekturen und Applikationen basieren auf Vertrauen. Es gibt allerdings Situationen, in denen Vertrauen nicht ausreicht – man muss die Sicherheit überprüfen.

Bild: Siemens

Als IT-Verantwortlicher hört man häufig Dinge wie: „Wir spielen Securitypatches umgehend ein.“, „Unsere Firewall ist 100% sicher, da kann niemand einbrechen.“, „Unser Produkt ist sicher, kein Hacker hat es jemals geknackt.“, „Wir sind auch im internen Netz sicher.“, „Wir wissen genau, welche Systeme in der DMZ stehen“ ...

Um hier aber Gewissheit zu haben, sollten regelmäßig Sicherheitsüberprüfungen durchgeführt werden. Dabei ist es mitunter von Vorteil, unabhängige Dritte hinzuzuziehen. Der externe Blick hilft eingefahrene Prozeduren oder scheinbare Selbstverständlichkeiten zu erkennen.

Ob Sie nun eine eigene Sicherheitsabteilung damit beschäftigen oder externe Dienstleister mit einer solchen Analyse beauftragen, grundsätzlich sollten gewisse Grundprinzipien bei der Analyse verfolgt werden:

- Das gesamte Testverfahren muss transparent gemacht werden.
- Es dürfen keine versteckten Annahmen gemacht werden, die etwa dazu führen, dass Bedrohungen unbetrachtet bleiben.
- Alle Bewertungen müssen begründet werden, um subjektive Einflüsse zu erkennen und so weit wie möglich zu vermeiden.
- Alle Schritte müssen so dokumentiert werden, dass sie später auch für andere nachvollziehbar sind. Ein derartiges Vorgehen erleichtert auch die Überarbeitung des daraus resultierenden IT-Sicherheitskonzeptes zu einem späteren Zeitpunkt.
- Der Aufwand für die Durchführung des Verfahrens sollte dem Wert der IT-Anwendungen und dem Bedrohungspotential der Organisation im Allgemeinen angemessen sein.

Ein Penetrationstest kann nur die Momentaufnahme eines bestimmten Untersuchungsgegenstandes zu einem bestimmten Zeitpunkt sein, mehr nicht. Die Sicherheitslöcher der verwendeten

Software sind bekannt, die Schwachstellen im Systemdesign sind bekannt, die bekannte Systemkonfiguration ist während der Untersuchung nicht geändert worden – eine Momentaufnahme. Ist dies für eine umfassende Bewertung ausreichend? Mit der geeigneten Vorgehensweise kann diese Momentaufnahme längeren Bestand haben als ein paar Stunden. Es ist möglich Aussagen zu erlangen, die den Status vielleicht über einige Wochen beschreiben. Aus diesem Grund ist es wichtig, derartige Tests regelmäßig zu wiederholen. Es ist auch sinnvoll, dass bei der Beauftragung eine konkrete Fragestellung für den Penetrationstest formuliert wird. Hier sollte festgelegt werden, was überprüft werden soll. Sind vertrauliche Daten einsehbar? Kann man in das interne Netz eindringen? Welche Informationen sind über die eigene Systemlandschaft im Internet „sichtbar"? Kann der Internetzugang beziehungsweise einzelne Komponenten lahmgelegt werden? Wurden die Intrusion Detection-Systeme sinnvoll implementiert? Werten die Administratoren die Protokolldateien regelmäßig aus?

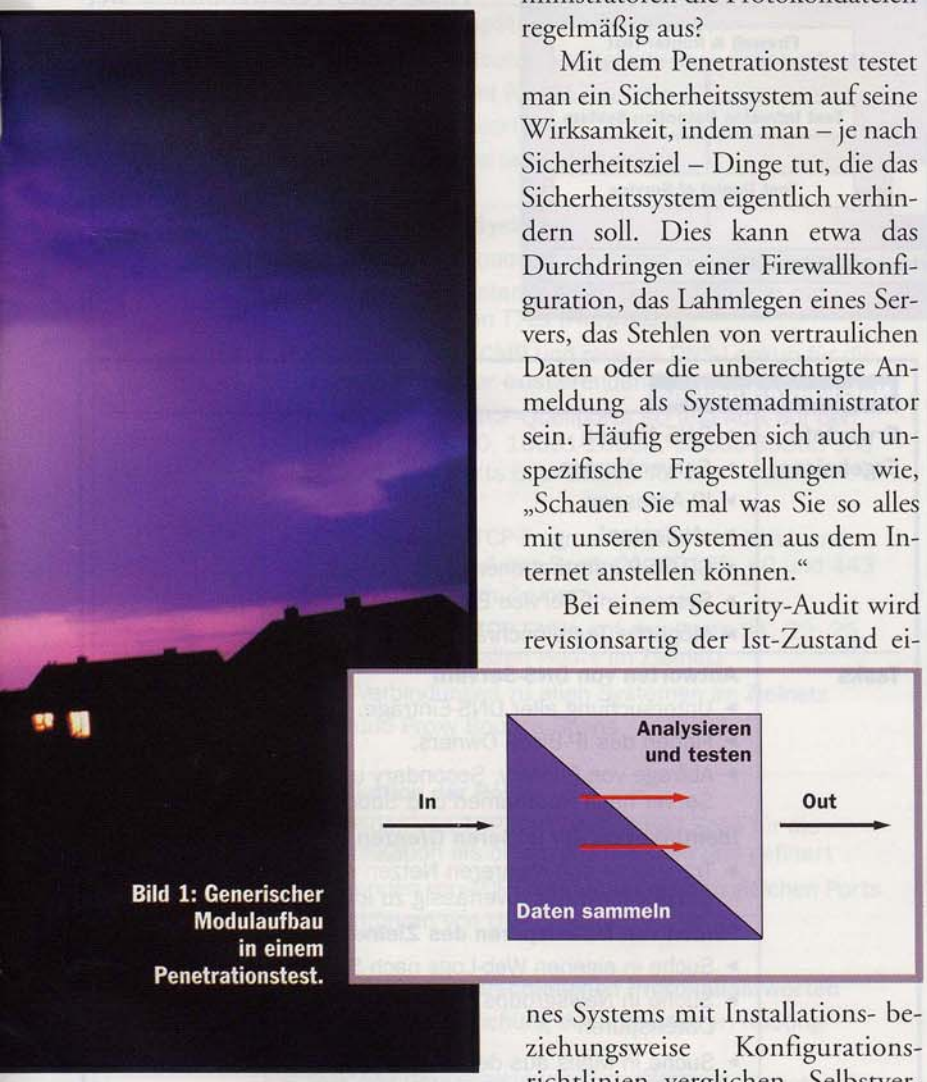

Bild 1: Generischer Modulaufbau in einem Penetrationstest.

Mit dem Penetrationstest testet man ein Sicherheitssystem auf seine Wirksamkeit, indem man – je nach Sicherheitsziel – Dinge tut, die das Sicherheitssystem eigentlich verhindern soll. Dies kann etwa das Durchdringen einer Firewallkonfiguration, das Lahmlegen eines Servers, das Stehlen von vertraulichen Daten oder die unberechtigte Anmeldung als Systemadministrator sein. Häufig ergeben sich auch unspezifische Fragestellungen wie, „Schauen Sie mal was Sie so alles mit unseren Systemen aus dem Internet anstellen können."

Bei einem Security-Audit wird revisionsartig der Ist-Zustand eines Systems mit Installations- beziehungsweise Konfigurationsrichtlinien verglichen. Selbstverständlich wird dabei gleichzeitig die Angemessenheit der Richtlinien überprüft. Wenn im Hause keine sinnvollen Anweisungen existieren, werden unter anderem generische Vorgaben und Checklisten herangezogen. Hierzu zählen das IT-Grundschutzhandbuch des BSI, international anerkannten Prüfkataloge, sowie Empfehlungen von Herstellern.

Bei einem sogenannten „Black Box"-Test werden alle Untersuchungen ohne jegliche Kenntnis des Untersuchungsgegenstandes durchgeführt. Bei diesem „Zero Knowledge-Test" ist das Ziel herauszubekommen, welche Erkenntnisse jemand X-beliebiges aus dem Internet erlangen kann, ohne in die konkrete Systemlandschaft eingeweiht zu sein.

Bei einem „White Box"-Test wird die Untersuchung mittels der Systemarchitektur, der Konfiguration oder des Quellcodes durchgeführt.

Generell lassen sich alle Untersuchungen nach der gleichen Methode durchführen. Die Analyse wird in Module aufgeteilt, die dann wiederum aus Tasks bestehen. In jedem Modul werden Daten gesammelt und sofort oder in einer weiteren Analysephase bewertet. Jedes Modul hat einen definierten Input und liefert ein zu erwartendes Ergebnis wie Bild 1 verdeutlicht.

Viele Tests sind Momentaufnahmen. Aus diesem Grund sind regelmäßige Wiederholungen notwendig.

Durch die Verbindung des Outputs eines Moduls mit dem Input für ein nächstes Modul entsteht eine Modulkette, an deren Ende ein Ergebnisbericht steht. Dieser Modulansatz ermöglicht auch die Trennung von Datensammlung und -analyse. Durch einen allzu formalen Ansatz darf aber die Kreativität der Testperson nicht eingeschränkt werden. Diese Kreativität des Testteams ist das eigentliche Potential, das die verschiedenen Anbieter voneinander unterscheidet und, neben der Erfahrung, als Alleinstellungsmerkmal gilt.

Hat ein Modul keinen Output als Ergebnis können folgende Ursachen dafür verantwortlich sein:

- Die Tasks wurden nicht richtig durchgeführt.
- Es wurden unpassende Tasks ausgewählt.
- Das darüber liegende Modul liefert kein Ergebnis.
- Das Ergebnis ist nicht richtig analysiert worden.

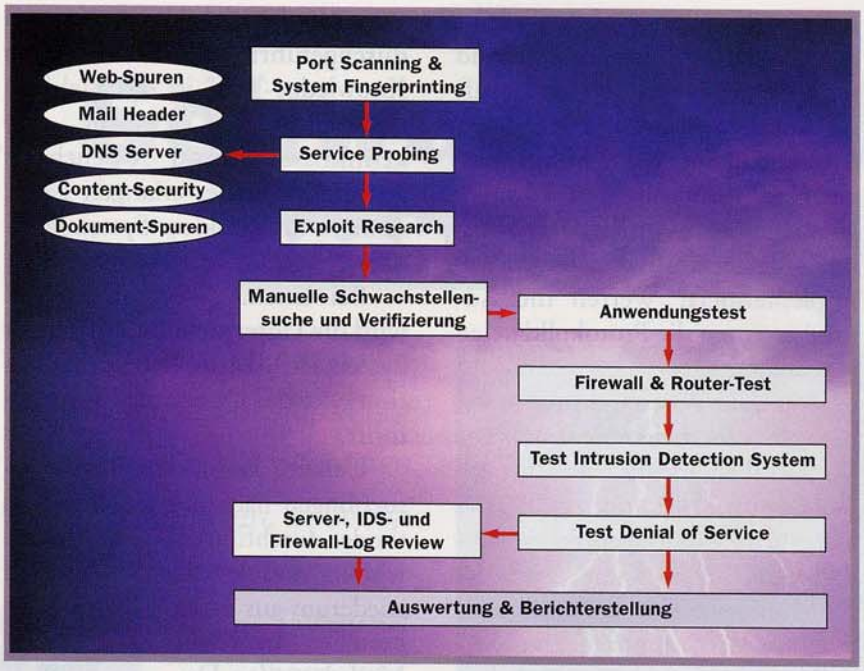

Bild 2: Beispielhafter Modulaufbau für einen Test aus dem Internet.

Hier müssen Qualitätssicherungsmaßnahmen greifen. Eine dieser Maßnahmen ist ein interner Review innerhalb des Testteams durch die Mitarbeiter, die am aktuellen Test nicht direkt beteiligt sind.

So können beispielsweise in einem Sicherheitstest des Interzugangs eines Unternehmens neben anderen die Module Netzwerkerkundung, Portscanning, Systemidentifikation und Service Probing enthalten sein. Als Input bekommt das Modul Netzwerkerkundung etwa den Namen der zu testenden Organisation, als Output muss eine zuverlässige Liste der zugehörigen IP-Netzblöcke geliefert werden. Dieser Output dient dann als Input für das Modul Portscanning. Dieses Modul liefert wiederum eine Liste der im Netz aktiven Systeme, der unterstützten Protokolle und offenen Ports. Diese Informationen dienen dem Modul Service Probing, die auf den offenen Ports laufenden Dienste, Serverversionen und Hersteller zu ermitteln. Das Modul Portscanning liefert aber auch den Input für das Modul Systemidentifikation, das als Output eine Aussage über die mögliche Funktion des Systems im Netzverbund liefern soll. Auf diesen Ergebnissen aufbauend können so in nachfolgenden Modulen spezielle Schwachstellen in Anwendungen und Sicherheitsarchitekturen getestet werden.

Zur Verdeutlichung der ersten Phase eines Penetrationstest werden nachfolgend Tasks und gelieferte Ergebnisse der Module Netzwerkerkundung und Portscanning beispielhaft aufgezeigt (siehe Tabelle 1 und 2).

Ein externer Penetrationstest sollte auf jeden Fall nur dann durchgeführt werden, wenn ein schriftlicher Auftrag zum Angriff

Netzwerkerkundung	
Erwartete Ergebnisse:	► DNS Namen ► Server Namen ► IP Adressen ► „Netzplan“ ► SP/ASP Informationen ► System und Service Eigentümer ► Mögliche Testeinschränkungen
Tasks	**Antworten von DNS-Servern** ► Untersuchung aller DNS-Einträge. ► Finden des IP-Block Owners. ► Abfrage von Primary, Secondary und ISP-DNS-Server nach Hostnamen und Subdomains
	Identifikation der äußeren Grenzen des Zielnetzes ► Traceroute von mehreren Netzen und Kontinenten um Borderrouter zuverlässig zu identifizieren.
	Finden von Datenspuren des Zielnetzes ► Suche in eigenen Web-Logs nach Spuren des Zielnetzes ► Suche in Newsgroups und Suchmaschinen nach Datenspuren ► Suche in Mails aus dem Zielnetz nach Datenspuren
	Informationslöcher ► Untersuchung der Ziel WWW-Server nach Quellcode, Anwendungsskripten und internen Links ► Untersuchung von nichtzustellbaren Mails nach internen IP-Adressen und Servernamen ► Suche in Newsgroups nach internen Informationen vom Zielsystem ► Suche in Stellenausschreibungen nach speziellen Skills im IT-Bereich „z.B. Suchen Systemadministrator mit Kenntnissen über Checkpoint-Firewall-1 auf Solaris“ ► Suche in P2P Netzen nach Verbindungen in Zielnetz
[...]	

Tabelle 1: Interessanterweise wird ein Systemadministrator von den Tasks des Moduls Netzwerkerkundung wenig bemerken, da hierbei nicht immer aktive Verbindungen zum Zielnetz aufgebaut werden.

Portscanning	
Erwartete Ergebnisse:	► offene, geschlossene und gefilterte IP-Adressen der aktiven Systeme ► Adressierung der internen Systeme ► Liste der entdeckten getunnelten und eingepackten Protokolle ► Liste der unterstützen Routingprotokolle ► Aktive Dienste ► „Netzplan“
Tasks	**Fehlerquellen für spätere Tasks erkunden** ► Überprüfen der Routen ins Zielnetz nach Paketverlusten ► Messen der Paket Round-Trip Time ► Messen der Antwortzeiten ► Messen der Höhe der Paketverluste ► [...]
	Identifikation der Systeme ► Sammeln der Broadcastantworten aus dem Zielnetz ► Erkunden der Systeme hinter Firewall/Router mit speziell gesetzten TTLs (Firewalking) ► Verwenden von ICMP und reverse DNS-Lookup für die Identifikation aller existierenden Systeme im Zielnetz ► Verwenden des TCP-Quellports 80 und ACK auf den Ports 3100-3150, 10001-10050, 33500-33550 und 50 zufälligen Ports über 35000 für alle Hosts im Zielnetz ► Verwenden von TCP-Fragmenten mit FIN, NULL und XMAS Scans1 auf den Ports 21, 22, 25, 80 und 443 bei allen Hosts im Zielnetz ► Verwenden von TCP SYNs auf den Ports 21, 22, 25, 80 und 443 bei allen Hosts im Zielnetz ► DNS-Verbindungen zu allen Systemen im Zielnetz ► FTP- und Proxy Bounce Scans ► [...]
	Identifikation der Ports ► erwenden von TCP SYN (Half Open) Scan für die Identifikation als offen, geschlossen und gefiltert ► Verwenden von TCP-Fragmenten auf den gleichen Ports ► Durchführen von UDP-Scans ► [...]
	Überprüfung der verschiedenen Protokollantworten ► Test und Untersuchung der unterstützen Routing-protokolle ► Test und Untersuchung der unterstützen IP-Protokolle ► Test und Untersuchung der unterstützen Tunnel und Crypto-Protokolle
	Untersuchung der Antwortpakete ► Untersuchung der TCP-Sequenznummern ► Untersuchung der Uptime (ermöglicht Aussagen über eingespielte Patches die ein Reboot benötigen)

Tabelle 2: Die weiteren Module können der Grafik (Bild 2) entnommen werden.

des Netzes vorliegt. In diesem Auftrag sollte das Ziel so genau wie möglich beschrieben sein. Zusätzlich sollte festgehalten werden, ob der Untersuchungsgegenstand durch einen Denial-of-Service Angriff lahmgelegt werden kann. Es sollte auch bedacht werden, den Auftrag auf etwaige Dritte, wie beispielsweise Hosting- oder Leitungsanbieter auszuweiten. Häufig werden zum Beispiel Chat- oder Community-Server zu Agenturen ausgelagert. Diese Server befinden sich zwar in einem anderen IP-Netz Block, gehören aber logisch zur eigenen Anwendungsumgebung. Werden hier Sicherheitsrisiken übersehen, kann sich das auf das gesamte System negativ auswirken.

Viele Anbieter von Sicherheitsanalysen und Penetrationstest erwähnen, dass sie zusätzlich zu den Standardtools selbstentwickelte Werkzeuge und Methoden zum Einsatz bringen. Erfahrungsgemäß sehen Berichte solcher Firmen dann aber leider aus wie ein Standardbericht von automatisierten Schwachstellenscannern wie etwa dem CyberCop Scanner oder dem Internet Security Scanner. Sicherlich ist es sinnvoll, diese mittlerweile zum Standard gehörenden Analysewerkzeuge einzusetzen. Die Frage ist aber, welche Tools verwendet ein potentieller Angreifer wirklich? Um ein möglichst realistisches Szenario zu erreichen, sollten zusätzlich die Tools eingesetzt werden, die sich in der Regel auf der Festplatte eines jeden Angreifers befinden und somit unbedingt in den gutgefüllten Werkzeugkasten eines fähigen Penetrationstesters gehören. Automatisierte Schwachstellenscanner sind häufig hilfreich, um in einem größeren Netzwerk eine erste Analyse durchzuführen. Man sollte sich aber niemals nur auf solche Tools allein verlassen. Die Gefahr der sogenannten False Positive Alarms (fälschlich gemeldete Schwachstellen) ist höher als man erwartet. Aus diesem Grund gehört die manuelle Verifizierung der vermuteten Schwachstelle unbedingt zu einer Untersuchung.

Automatische Schwachstellenscanner alleine reichen heutzutage nicht mehr aus.

Alexander Geschonneck
geschonneck@hisolutions.com