



Introduzione al servizio

Checked by SecureGate v.1.0 – 2005

Sommario

1	CHECKED BY SECUREGATE	3
2	MODULI	3
2.1	CONTROLLO PERIMETRALE – ETHICAL HACKING –	3
2.2	CONTROLLO PERIMETRALE – APPLICATION HACKING –	4
2.3	CONTROLLO PERIMETRALE – VULNERABILITY ASSESSMENT –	4
2.4	CONTROLLO DELLA RETE INTERNA – ETHICAL HACKING –	5
2.5	CONTROLLO DELLA RETE INTERNA – VULNERABILITY ASSESSMENT –	6
2.6	MANUTENZIONE PREVENTIVA	7
2.7	SECURITY ADVISOR	7
2.8	HELP DESK	7
2.9	ANALISI DEL RISCHIO	7
3	CHECKED BY SECUREGATE – COPERTURA TEMPORALE	8
4	LOGO SECUREGATE CHECKED	9
5	LIMITAZIONI DEL SERVIZIO	9
6	DOCUMENTAZIONE	10
7	ETHICAL HACKING – INTRODUZIONE	11
8	ETHICAL HACKING – APPROFONDIMENTI	11
9	DESCRIZIONE DEL SERVIZIO	11
9.1	TEST DALL'ESTERNO	12
9.2	TEST DALL'INTERNO	12
9.3	SCAN TELEFONICO	13
10	VANTAGGI DEL SERVIZIO DI ETHICAL HACKING	14
11	MODALITÀ DI INTERVENTO	14
12	METODOLOGIE	15
12.1	ANALISI NON INVASIVA	15
12.1.1	Footprinting	15
12.1.2	Scanning	15
12.2	ANALISI INVASIVA	15
12.2.1	Scanning tools	16
12.2.2	Enumeration	16
12.3	ATTACCO	16
12.3.1	Gaining access	16
12.3.2	Escalating privileges	16
12.4	CONSOLIDAMENTO	17
12.4.1	Pilfering	17
12.4.2	Covering traces and creating backdoors	17
13	ETHICAL HACKING APPLICATIVO	17
13.1	CLASSI DI ATTACCO APPLICATIVO	18
13.1.1	Parameter tampering	18
13.1.2	Hidden field manipulation	18
13.1.3	Backdoors & debug options	18
13.1.4	Cookie poisoning	18
13.1.5	Forceful browsing	19
13.1.6	Cross-Site scripting	19
13.1.7	Buffer overflow	19
13.1.8	Known vulnerabilities	19
13.2	TIPOLOGIE DI ATTACCO	20
14	ETHICAL HACKING – CONCLUSIONI	24
14.1	TOOL AUTOMATICI	24
14.2	INTRUSIONE DEL BLACK HAT	24

1 Checked by SecureGate

Il servizio Checked by SecureGate risponde alla crescente esigenza, da parte delle aziende, di ottenere sempre più alti livelli di sicurezza nelle proprie infrastrutture informatiche. Questo servizio comprende una serie di attività di assessment, monitoring e consulenza, finalizzate ad identificare le criticità dei sistemi aziendali ed a mitigarle.

Pur essendo un servizio modulare, pensato per adattarsi alle più diverse realtà aziendali, Checked by SecureGate deve essere considerato come un unico strumento volto a controllare e migliorare lo stato di sicurezza della rete e dei flussi delle informazioni nell'azienda.

Per ottenere questo obiettivo, è necessario modificare l'approccio comunemente adottato per affrontare le problematiche di sicurezza, utilizzando un metodo che *attivamente* identifichi i problemi e li risolva prima che questi diventino rischi. Questo risultato non si ottiene quindi con un controllo saltuario dell'infrastruttura, ma con approccio sistematico e rigoroso, realizzato in modo competente. Solo un controllo continuo, quindi, può garantire un maggior livello di sicurezza.

Il problema principale delle infrastrutture di sicurezza, come spesso avviene per i processi aziendali, è l'obsolescenza. Un componente dell'infrastruttura, se abbandonato a se stesso, diventa presto un punto debole che metterà a rischio anche gli altri elementi.

2 Moduli

Checked by SecureGate è composto da più elementi, i quali si completano analizzando i diversi aspetti tecnologici ed organizzativi aziendali. Ogni modulo, pur se distinto dagli altri, aggiunge un'informazione che completa l'immagine della situazione di sicurezza dell'infrastruttura. Quando alcuni di questi moduli non sono applicabili alla particolare situazione dell'Azienda, possono essere esclusi, senza pregiudicare il meccanismo di controllo.

Di seguito analizzeremo i moduli attualmente previsti.

2.1 Controllo perimetrale – *Ethical Hacking* –

Il primo modulo prevede l'assessment della rete aziendale esposta ad Internet. Questo controllo dev'essere effettuato da remoto, per simulare in modo completo il comportamento di un "*hacker*", intenzionato a violare le difese dell'Azienda. In questo caso, le attività effettuate da SecureGate sono svolte da tecnici specializzati, con una lunga esperienza nel campo della sicurezza informatica ed in particolare in tecniche di *hacking*. Questi specialisti hanno l'obiettivo di verificare il livello di protezione dell'infrastruttura esposta ad Internet, utilizzando una metodologia ben definita. Prima di iniziare le attività di controllo perimetrale (*Ethical Hacking*), è necessario un incontro tecnico con il Cliente, per stabilire il livello di profondità di questo servizio, che può variare dal semplice controllo all'attacco limitato, fino ad arrivare al tentativo di intrusione.

Questo modulo effettua un controllo generalmente molto approfondito, quindi la sua frequenza è quantificabile in 3 - 4 interventi nell'arco dell'anno,

in conformità anche alle direttive stabilite con il Cliente. Per una descrizione dettagliata di questo servizio, ed approfondimenti sulla metodologia, si faccia riferimento ai capitoli seguenti (Cap. 7 e successivi).

A valle di ogni test di Ethical Hacking è prevista la stesura di un documento riassuntivo dei problemi riscontrati e, quando possibile, delle relative soluzioni. Questo *report* sarà discusso con la divisione tecnica del Cliente, direttamente dal consulente di SecureGate che ha effettuato il test e dal project manager di riferimento. In questo modo, anche la risoluzione dei problemi riscontrati è immediata.

È prevista infine la possibilità di affiancare ai tecnici di SecureGate, se richiesto, i tecnici del Cliente in modo da consentire un trasferimento di conoscenze, e di verificare in tempo reale le tecniche di attacco utilizzate.

2.2 Controllo perimetrale – *Application Hacking* –

Lo strato di *networking* è sicuramente una componente importante dell'infrastruttura aziendale, ma con lo sviluppo delle tecnologie e con l'aumentare della sicurezza del *canale di trasmissione*, si sono sempre più diffuse le applicazioni (esposte ad Internet) in grado di mettere a disposizione del pubblico informazioni aziendali. È il caso dei siti di trading on line, di home banking ma anche di commercio elettronico e siti di consultazione.

In presenza di questo genere di applicazioni, aumenta notevolmente il livello di rischio per l'Azienda; questo accade in quanto le applicazioni WEB costituiscono – di fatto – una porta verso l'interno dell'Azienda stessa. Se non sono progettate con particolare attenzione alle problematiche di sicurezza, infatti, le applicazioni WEB possono concedere a potenziali attaccanti accesso a dati riservati o addirittura, nei casi peggiori, la possibilità di effettuare operazioni non autorizzate.

Per valutare l'effettiva presenza delle vulnerabilità legate alle applicazioni, SecureGate è stata tra i primi a proporre attività di Ethical Hacking, focalizzate esclusivamente sulle applicazioni (*Application Hacking*). Queste prendono in considerazione in modo particolare il server che ospita l'applicazione e l'applicazione stessa. Maggiori approfondimenti sulla metodologia impiegata per questo genere di operazioni, sono presenti nei prossimi capitoli (Cap. 7 e successivi).

Al termine dei test effettuati sull'applicazione, SecureGate produce un documento di report nel quale sono indicate le eventuali problematiche riscontrate, e le possibili soluzioni a questi problemi. Come nel caso precedente, il report è illustrato al Cliente, direttamente dallo specialista che ha effettuato il test.

2.3 Controllo perimetrale – *Vulnerability Assessment* –

Il terzo modulo perimetrale è rappresentato dal servizio di *Vulnerability Assessment*. Questo tipo di controllo è simile al precedente, ma non prevede il coinvolgimento diretto di personale di SecureGate, in quanto è effettuato mediante l'utilizzo di strumenti automatici.

In particolare, SecureGate utilizza uno strumento proprietario, SGBBox, in grado di effettuare scan automatici ad un insieme di *host* esposti ad Internet. SGBBox (<https://www.sgbox.it>) è uno strumento che integra alcuni tra i più noti

tool di scan automatico, per produrre un report delle vulnerabilità riscontrate durante l'assessment di un determinato *host*.

SGBBox ha numerosi punti di forza, che lo rendono uno strumento particolarmente adatto ad effettuare un controllo esteso e continuato sui servizi. Lo scopo principale di SGBBox, infatti, è di fornire all'utilizzatore una serie di report effettivamente fruibili. Se i comuni tool di scan automatico producono report di difficile interpretazione e spesso anche di difficile consultazione, SGBBox produce report differenziati, a seconda della funzione aziendale a cui sono rivolti. Per questo motivo, sono stati progettati report specifici per il management, che graficamente riassumono lo stato di sicurezza dell'infrastruttura, e report sempre più dettagliati rivolti ad un'utenza tecnica, che dovrà poi verificare lo stato dei server. I report di SGBBox possono essere incrementali e di trend: è possibile infatti confrontare solo due test, per verificare l'effettivo miglioramento della situazione tra il primo ed il secondo, o invece confrontare più test, per visualizzare il trend del livello di sicurezza dell'infrastruttura testata.

Un ulteriore punto di forza di SGBBox è il test proattivo dell'infrastruttura, vale a dire la modalità denominata "*Sentinel mode*". Con questa modalità SGBBox, a valle di uno scan integrale, verifica la presenza di tutte e sole le nuove vulnerabilità, di volta in volta pubblicate. In questo modo è possibile verificare lo stato di sicurezza di ogni singolo server, fino a quattro volte al giorno, e ricevere il risultato della verifica se questa è andata a buon fine.

Il controllo perimetrale mediante Vulnerability Assessment, proprio grazie alle funzionalità messe a disposizione da SGBBox, diventa quindi un controllo *continuo*. Le verifiche, svolte al massimo ogni sei ore, assicurano una copertura costante ed una rapida risposta in caso di rilevazione di vulnerabilità.

La gestione dei report è demandata ad SGBBox, che consente un accesso diretto ed interattivo ai risultati dei test. L'utente incaricato della gestione dei report, infatti, ha la possibilità di visionare i test, effettuare confronti o grafici di trend, salvare le configurazioni selezionate (*template*) per un uso futuro, o stampare i report ottenuti per poi distribuirli alle figure predisposte. Ulteriori dettagli sono disponibili nel documento allegato SGBBox.pdf.

2.4 Controllo della rete interna – *Ethical Hacking* –

La maggior parte degli attacchi informatici proviene dall'interno della rete aziendale. Per questo motivo è importante mantenere sotto controllo, oltre al perimetro della rete, anche i servizi presenti al suo interno.

SecureGate, grazie alle proprie conoscenze e all'esperienza maturata in anni di attività consulenziale, è in grado di offrire un servizio di *Ethical Hacking* ad alto livello anche della rete interna, dal quale vengono identificati i punti di debolezza dell'infrastruttura. La struttura del servizio di Ethical Hacking della rete interna è simile a quello descritto nel capitolo relativo al test perimetrale.

In generale, il servizio di Ethical Hacking svolto all'interno della rete aziendale, è diviso in tre fasi:

1. analisi dell'infrastruttura ed attacco "*blind*"
2. riunione con il Cliente
3. analisi dei punti considerati critici

La prima fase simula al meglio l'attacco effettuato da un utente senza particolari privilegi che, senza conoscere la topologia della rete cerchi di ottenere informazioni o di assicurarsi il controllo dei servizi. In questa fase, i tecnici di SecureGate non conoscono la rete e cercano di raccogliere il maggior numero di informazioni.

La seconda fase, dopo una prima raccolta dati ed una serie di attacchi realizzati, prevede una riunione con il Cliente, in modo da focalizzare un secondo attacco sugli host ritenuti particolarmente importanti. Se questi non sono già stati analizzati nella prima fase, si procede con l'ultima parte dell'assessment che vede appunto l'analisi dei punti "critici".

Durante la prima fase dell'analisi sono presi in considerazione tutti i diversi segmenti di rete raggiungibili, mentre nell'ultima parte sarà possibile effettuare i test anche da altri punti della rete interna.

A valle di questi test, oltre al report di vulnerabilità che elenca gli eventuali problemi riscontrati, è fornito un documento con una rappresentazione grafica della rete analizzata. Quest'ultimo documento sarà la base per un'analisi, da perfezionare in accordo con il Cliente, finalizzata ad ottenere un layout di rete orientato alla sicurezza. Lo scopo ultimo è quello di, se ritenuto necessario, modificare la struttura della rete per renderla ulteriormente sicura.

Come per l'analisi perimetrale, anche in questo caso è possibile che i tecnici del Cliente affianchino i tecnici di SecureGate. In modo particolare nella terza fase dell'assessment, è comunque prevista una forte interazione con il Cliente.

2.5 Controllo della rete interna – *Vulnerability Assessment* –

Un ulteriore modulo, di fondamentale importanza, è quello che prevede controlli periodici sulla rete interna. Come nel caso indicato in precedenza circa i Vulnerability Assessment per la rete perimetrale, anche per la rete interna è necessario effettuare controlli ripetuti per prevenire possibili attacchi. Anche in questo caso SecureGate utilizza SGBBox, per affrontare questa problematica. SGBBox, infatti, può essere utilizzato per effettuare i test di vulnerabilità anche nella rete interna aziendale; SecureGate in questo caso fornisce al Cliente un'appliance preinstallata che permette di gestire i test all'interno della rete aziendale. In questo caso è possibile organizzare in asset gli host da testare, e temporizzare i test, in modo da effettuarli ad orari prefissati, solo in determinati giorni.

Come nel caso precedente, è possibile attivare il *Sentinel Mode* sugli asset di interesse, per ottenere un altissimo livello di controllo.

La scelta dei punti da analizzare e la creazione degli asset è generalmente effettuata da SecureGate in accordo con il Cliente; il Cliente, in questo caso, ha il completo controllo su SGBBox, ed ha la possibilità di modificare aggiungere o eliminare gli asset stabiliti.

Come nel caso precedente, SGBBox permetterà di effettuare test di trend, che indicheranno con estrema chiarezza l'evolversi del livello di sicurezza nel tempo, e di verificare l'effettiva utilità degli interventi svolti.

2.6 Manutenzione preventiva

Il servizio Checked by SecureGate prevede una serie di interventi programmati, da parte di tecnici di SecureGate, per l'installazione di *patch* ed *upgrade* sui sistemi del Cliente. Queste attività sono di primaria importanza per mantenere aggiornati i sistemi, e per ottenere un migliore livello di sicurezza.

Gli interventi programmati, sono generalmente utilizzati anche per effettuare verifiche sulla correttezza delle configurazioni, e per l'analisi periodica dei log provenienti dagli apparati di sicurezza.

Questa serie di interventi programmati, stabilita in accordo con il Cliente, è funzione della dimensione, tipologia e criticità dei server utilizzati.

2.7 Security Advisor

SecureGate ha identificato una figura professionale, quella del *Security Advisor*, con il compito di seguire i Clienti su determinate problematiche. In particolare, il *Security Advisor* conosce in profondità l'infrastruttura del Cliente, ed è in grado di suggerire, a fronte delle verifiche effettuate, quali soluzioni adottare per porre rimedio ai problemi emersi.

Il *Security Advisor* è un riferimento affidabile per il Cliente, per tutto quanto riguarda la *computer security*. Questi, forte delle conoscenze specifiche sulla realtà del Cliente, è in grado di proporre nuove soluzioni *software/hardware* per migliorare il livello di sicurezza, ma anche di fornire un valido supporto tecnico per modificare le configurazioni o il layout della rete.

Compito del *Security Advisor* è di creare un canale prioritario di comunicazione tra SecureGate ed il Cliente, divenendo il punto di riferimento per le problematiche di sicurezza.

A fronte di un attacco o di un evento critico, il *Security Advisor* è in grado di analizzare la situazione e proporre attivamente delle contromisure.

Il *Security Advisor* supporta il Cliente nella soluzione dei *security incidents* e, analizzando i log degli attacchi, propone misure tecniche e procedurali per mitigare il rischio.

Produce e commenta con il Cliente il report delle avvenute attività, e insieme al Cliente definisce un quadro globale della situazione, indirizzando eventuali problemi rimasti aperti.

2.8 Help desk

Il servizio Checked by SecureGate comprende un modulo di *help desk*, che consente al Cliente di accedere direttamente alle competenze di SecureGate.

Mediante il modulo di help desk, il Cliente può consultare i tecnici di SecureGate per ottenere informazioni per la risoluzione di problemi ordinari, o di carattere più specialistico sui componenti di sicurezza installati. Questo servizio è disponibile sia telefonicamente, che via e-mail.

2.9 Analisi del rischio

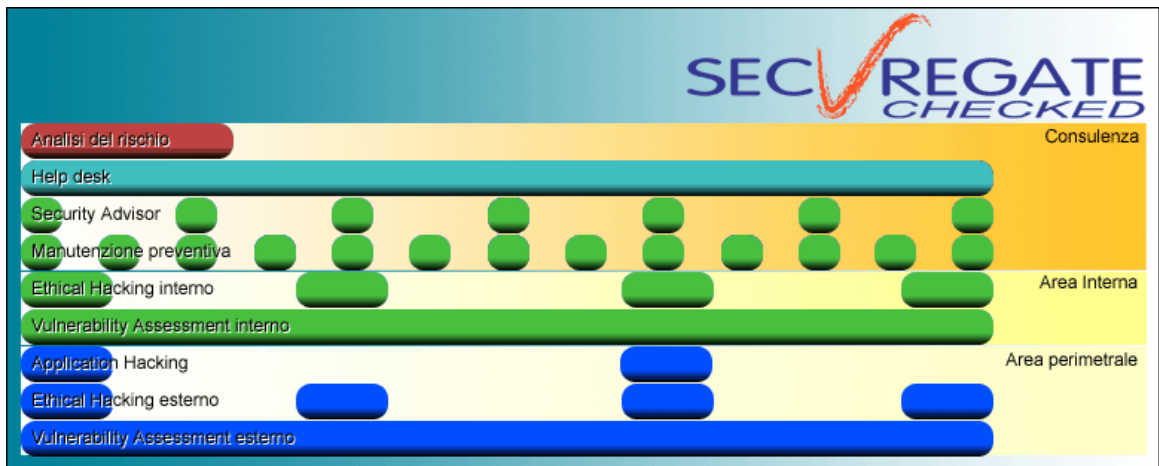
I moduli esaminati finora sono rivolti principalmente alla gestione della sicurezza da un punto di vista prettamente tecnologico. SecureGate è in grado di analizzare anche la componente di rischio legata alle aree gestionali ed organizzative dell'Azienda.

Con il servizio di analisi del rischio, SecureGate è in grado di erogare consulenza su aree di vulnerabilità, di non conformità agli standard di sicurezza, sull'adeguatezza del piano di sicurezza e sulla verifica delle security policy. SecureGate è in grado di redigere il *business continuity plan*, ed effettuare analisi per l'identificazione delle risorse critiche all'interno dell'Azienda, fino a preparare l'Azienda stessa alla certificazione BS7799.

L'analisi del rischio è quindi un componente prioritario all'interno del servizio Checked by SecureGate.

3 Checked by SecureGate – copertura temporale

In figura sono rappresentati graficamente i moduli del servizio Checked by SecureGate ed il loro posizionamento standard nell'arco temporale del servizio stesso.



Per ognuna delle tre aree (interna, perimetrale, consulenza) i moduli si sovrappongono per mantenere sempre alto il livello di controllo.

Nel dettaglio, in quella denominata *Area perimetrale*, è effettuata una verifica costante con strumenti automatici (SGBBox), che si preoccupano di testare con cadenza giornaliera la presenza di nuove vulnerabilità sull'asset esterno. Parallelamente, con frequenza inferiore, sono effettuati degli interventi di *Ethical Hacking* finalizzati, con l'ausilio di personale specializzato, ad evidenziare eventuali problematiche che non possono essere rilevate da sistemi automatici. Dato che le applicazioni, di norma, sono modificate raramente dopo la messa in produzione, il modulo di *Application Hacking* è riportato con frequenza ancora inferiore. La cadenza di questo test può variare, in accordo con le policy del Cliente.

Nell'*Area Interna* sono riportati i moduli di *Vulnerability Assessment*, e di *Ethical Hacking*. In modo del tutto analogo al precedente il vulnerability assessment effettuato con SGBBox garantisce un controllo costante, mentre il modulo di ethical hacking verifica le zone critiche, con cadenza inferiore.

L'area di *Consulenza*, infine, supporta il Cliente con frequenti interventi di manutenzione preventiva, e con interventi del *Security Advisor*, volti a mantenere aggiornata l'infrastruttura. La presenza costante, in quest'area è data dall'*Help Desk*, che fornisce al Cliente un punto di riferimento per la risoluzione dei problemi di ordinaria amministrazione.

L'analisi del rischio, effettuata una sola volta nell'arco del periodo di copertura di Checked by SecureGate, indirizza le problematiche organizzative e di compliance alle policy aziendali.

4 Logo SecureGate Checked

SecureGate, una volta stabilito il livello di controllo ed i moduli da adottare per le singole esigenze del Cliente, inizierà ad effettuare i servizi descritti nei capitoli precedenti.

Se il Cliente lo desidera, sarà possibile aggiungere al proprio sito il logo SecureGate Checked. Il logo è utilizzabile per tutto il tempo per cui il servizio è attivo. In questo caso, il Cliente potrà utilizzare il logo SecureGate Checked a patto che siano verificati da SecureGate i requisiti minimi di sicurezza previsti per questo servizio.

5 Limitazioni del servizio

Il servizio Checked by SecureGate garantisce al Cliente un controllo attento e continuativo sulle infrastrutture che il Cliente stesso deciderà di affidare a SecureGate per questo genere di attività.

Il *database* delle vulnerabilità ricercate, nell'ambito del controllo automatico (*SGBox*), è aggiornato con frequenza quotidiana (ogni 6 ore).

Il servizio Checked by SecureGate non può, invece, garantire l'assoluta inviolabilità dei sistemi oggetto di protezione, in quanto questo livello di sicurezza *non può essere garantito da alcun ente*. Il controllo perimetrale ed interno, eseguito con strumenti automatici (*SGBox*) o manualmente (*ethical hacking*), ricerca le vulnerabilità note, ma non può effettuare controlli su vulnerabilità non note.

6 Documentazione

In questa sezione è elencata la documentazione prevista per ogni modulo.

- **Controllo perimetrale (*Ethical Hacking*)**
 - Report di termine lavori. Il report è presentato e discusso con il Cliente.
- **Controllo perimetrale (*Application Hacking*)**
 - Report di termine lavori. Il report è presentato e discusso con il Cliente.
- **Controllo perimetrale (*Vulnerability Assessment*)**
 - I report sono configurabili e prodotti on-demand da SGBBox.
 - Report “*executive*”
 - Report “*summary*”
 - Report “*technical*”
 - Report differenziali e di trend
 - Opzionalmente i report sono presentati al Cliente
- **Controllo della rete interna (*Ethical Hacking*)**
 - Report di termine lavori. Il report è presentato e discusso con il Cliente.
- **Controllo della rete interna (*Vulnerability Assessment*)**
 - I report sono configurabili e prodotti on-demand da SGBBox.
 - Report “*executive*”
 - Report “*summary*”
 - Report “*technical*”
 - Report differenziali e di trend
 - Opzionalmente i report sono presentati al Cliente
- **Manutenzione preventiva e *Security Advisor***
 - Report di intervento riassuntivi delle attività svolte
- **Help desk**
 - Comunicazione di avvenuta presa in carico del problema (e-mail)
 - Comunicazione di chiusura “ticket” (e-mail)
 - Eventuali altre comunicazioni al Cliente (e-mail)
- **Risk Assessment**
 - Obiettivi del progetto, scelta della metodologia
 - Analisi degli asset: una valutazione di sistemi, dati, policy e persone coinvolte nel progetto
 - Studio dello spettro delle minacce applicabili agli asset: screening e analisi di sensitività
 - Vulnerability assessment (integrato nel RA)
 - Valutazione dei controlli esistenti
 - Calcolo del fattore di rischio
 - Risk Management: metodologie e strumenti per ridurre o trasferire il rischio
 - Analisi costo-Beneficio
 - Piano di implementazione e delle misure di riduzione del rischio
 - Mantenimento e ottimizzazione del Piano

7 Ethical Hacking – introduzione

Nei prossimi capitoli, sarà descritto il servizio di *Ethical Hacking*, sia per la parte perimetrale (esterno) che per quella interna; sarà inoltre illustrato il test *telefonico*, che in determinate situazioni può essere affiancato ai primi due. SecureGate, grazie alla lunga esperienza in questo ambito, è in grado di proporre al Cliente un servizio di Ethical Hacking di alta qualità.

8 Ethical Hacking – approfondimenti

Il *Ethical Hacking* permette di evidenziare i punti deboli di un'infrastruttura di rete e quindi consentire uno studio della rete stessa, volto a migliorare il livello di sicurezza globale.

Il metodo più efficace, infatti, per evidenziare i problemi di sicurezza ed i malfunzionamenti di una rete è proprio quello di simulare l'attacco da parte di un *hacker* intenzionato a prenderne il controllo.

È complesso riassumere le tecniche utilizzate da un esperto di sicurezza nell'affrontare questo genere di problematiche, in quanto queste non sono sempre uguali. Le vulnerabilità di una rete possono essere diverse e causate da differenti fattori: l'errata configurazione o il malfunzionamento temporaneo di un apparato, di un software o un bug applicativo, possono mettere l'*hacker* in condizione di acquisire privilegi all'interno del network.

In generale si può riassumere il Ethical Hacking in tre fasi:

1. Studio della situazione
2. Individuazione delle vulnerabilità
3. Utilizzo delle vulnerabilità

Nei prossimi capitoli si descriveranno più nel dettaglio queste fasi.

Al termine del test, verrà consegnato un report contenente:

1. L'elenco dei sistemi testati, e le vulnerabilità riscontrate per ciascuno di essi secondo una classificazione del livello di rischio (basso, medio, alto).
2. Per ogni debolezza riscontrata verrà descritto l'intervento atto a risolverla.
3. Nel caso in cui la vulnerabilità dovesse essere il risultato di più operazioni, verrà descritto nel dettaglio il modo per riprodurla.

9 Descrizione del servizio

Il *Ethical Hacking* è costituito da tre principali elementi:

1. Test dall'esterno (Internet)
2. Test dall'interno (Intranet)
3. Scan telefonico

9.1 Test dall'esterno

Il primo dei tre test è rivolto verso le difese *esterne* del network aziendale, cioè quelle che proteggono la rete da Internet o che permettono l'utilizzo della Rete dall'interno. Generalmente il test viene eseguito in modalità *blind*, cioè non vengono forniti indizi sulla topologia della rete e sulla natura delle protezioni poste tra Intranet ed Internet. Se invece si desidera avere un più dettagliato quadro delle vulnerabilità della propria rete, è possibile fornire qualche indizio (discusso in separata sede, prima dell'inizio del test) per sottoporre a maggiore stress le proprie protezioni.

L'attacco, che viene portato da un punto casuale della Rete, ha come finalità il controllo delle seguenti entità:

1. Firewall
2. Web
3. Mail server
4. Servizi pubblici

e, più in generale, tutte le macchine appartenenti al network con un indirizzo pubblico, raggiungibili dall'esterno.

In questa occasione viene verificata la presenza delle più recenti vulnerabilità applicative sui servizi posti sulle macchine target.

Inoltre viene verificata la presenza di servizi soggetti a *DoS*, Denial Of Service, che, pur non mettendo a repentaglio l'integrità dei dati presenti sull'Intranet, possono causare un'interruzione del servizio e creare ingenti danni economici e di immagine all'Azienda.

9.2 Test dall'interno

Secondo studi effettuati dal C.E.R.T.¹ risulta che la maggior parte dei tentativi di attacco nei confronti di sistemi informatici, proviene proprio dalla rete aziendale interna. Per questo motivo, è importante verificare la robustezza dei sistemi e degli apparati presenti sul network (errata configurazione di condivisioni di risorse, password banali, versioni non aggiornate dei servizi ecc.).

Il *Ethical Hacking* interno, mette in evidenza tutte queste problematiche e permette di prendere adeguate contromisure per la protezione dei sistemi.

Contrariamente a quanto detto per il test esterno, in questo caso è necessario fornire il maggior numero possibile di informazioni a chi lo esegue, in modo da massimizzare l'efficacia del servizio. In particolare, è necessaria una descrizione della topologia della rete, gli indirizzi e le tipologie dei server da sottoporre a test, gli eventuali router o altri apparati di rete.

Come nel primo caso verranno controllati i servizi attivi, le vulnerabilità dei pacchetti applicativi e la vulnerabilità ai DoS (Denial of Service)

¹ Computer Emergency Response Team (<http://www.cert.org>)

10 Vantaggi del servizio di Ethical Hacking

L'Ethical Hacking offre l'opportunità di mettere alla prova il proprio network, utilizzando le stesse tecniche di attacco che gli hacker usano su Internet.

L'Ethical Hacking non viene effettuato utilizzando programmi di scanning automatico della rete (es. Internet Scanner), se non durante la fase di raccolta di informazioni, ma utilizzando le conoscenze degli esperti del settore, aggiornati sulle ultime novità in questo ambito. I tool automatici utilizzati all'inizio del test (si veda nei capitoli successivi) forniscono importanti indicazioni ed una serie di linee guida ma, come più avanti avremo modo di spiegare, non sono sufficienti per portare a termine il test. Il probe, infatti, ha la finalità di sfruttare le vulnerabilità riscontrate cercando di scoprirne altre. Sicurezza dinamica, quindi, e non legata a software che opera secondo schemi prefissati.

Con questo test, è possibile simulare in tutto e per tutto il comportamento di un hacker, senza subire le conseguenze di un attacco.

Al termine dell' Ethical Hacking, viene prodotto un report che descrive in modo dettagliato tutte le debolezze riscontrate, mettendone in risalto la criticità ed assegnando loro un livello di rischio (basso, medio, alto). Inoltre, il report contiene una serie di suggerimenti su come porre rimedio alle eventuali vulnerabilità riscontrate (applicazione di patch, correzione delle configurazioni, ecc.) che permettono di elevare il livello di sicurezza globale della rete.

11 Modalità di Intervento

È consigliabile evitare di mettere al corrente dell' Ethical Hacking il target, in quanto uno degli obiettivi del probe è anche quello di valutare i tempi di reazione di chi lo commissiona. Avvertendo preventivamente il personale tecnico, si potrebbe inficiare il risultato del test.

In determinate situazioni, però, è necessaria una collaborazione tra SecureGate ed i tecnici del Cliente, ad esempio se il servizio fornito è particolarmente critico, o deve garantire assoluta continuità. In questo caso, viene fornito al cliente un piano delle tempistiche degli interventi e le fasce orarie in cui SecureGate effettuerà i test.

L'Ethical Hacking dall'esterno potrà essere effettuato anche da una linea in dialup, simulando il comportamento di un hacker. In questo modo sarà ancora più complesso rendersi conto dell'avvenuto attacco, o ricondurre l'ultimo attacco eventualmente registrato ad uno precedente.

L'Ethical Hacking della rete interna, invece, dovrà essere eseguito dall'interno della rete locale, quindi negli uffici del Cliente.

Al termine dell'Ethical Hacking esterno, verrà preparato il report analogo a quello del test interno. Sarà quindi possibile, tra il primo ed il secondo intervento, procedere a migliorie e fix sulle vulnerabilità riscontrate.

12 Metodologie

Descriveremo ora nel dettaglio² l'approccio utilizzato per portare a termine l'attacco nei confronti del target. L'attacco è suddiviso per semplicità in sei stadi (footprinting, scanning, scanning tools, enumeration, attacco e consolidamento) e per ognuno è fornita un'indicazione di quali passi sono portati a compimento.

I sei stadi del processo sono raggruppati in fasi identificate con "analisi non invasiva" ed "analisi invasiva". La prima fase fornisce le informazioni, la seconda porta a compimento il test.

12.1 Analisi non invasiva

In questo momento è necessario raccogliere informazioni sul proprio obiettivo, per ottenere una buona base di partenza.

12.1.1 *Footprinting*

Questa fase ha lo scopo di raccogliere il maggior numero di informazioni sull'obiettivo che si intende attaccare senza entrare in contatto con l'obiettivo stesso, ovvero effettuando una cosiddetta "analisi non invasiva". In particolare in questa fase è importante determinare: *domini, blocchi di rete e gli indirizzi ip* dei sistemi direttamente collegati ad internet. Gli strumenti utilizzati sono: Search Engine, server whois, database Arin ed interrogazioni ai DNS. In questa fase si raccolgono anche utili informazioni per effettuare eventuali attacchi di tipo *social engineering*. Questa raccolta di informazioni è assolutamente legale e può essere svolta da chiunque.

12.1.2 *Scanning*

Scopo dello scanning è quello di ottenere una mappa il più dettagliata possibile del sistema da attaccare; ciò significa acquisire informazioni su quali ip dei blocchi di rete trovati nella fase precedente siano effettivamente raggiungibili dall'esterno (IP discovery) e, relativamente a tali IP, scoprire che servizi abbiano attivi (Tcp/udp port scan) e che sistemi operativi possiedano. Gli strumenti utilizzati in questa fase sono: interrogazioni ICMP (gping, fping, ecc.), la scansione delle porte TCP e UDP (strobe, netcat, nmap, ecc.) e fingerprint dello stack (nmap, queso).

12.2 Analisi invasiva

Ottenute le informazioni necessarie, si inizia l'attacco vero e proprio.

² Si faccia riferimento anche a "*Hacking exposed*" di McClure, Scambray, Kurtz.

12.2.1 *Scanning tools*

Per agevolare il lavoro di test in questa fase dell'analisi vengono utilizzati tool automatici che forniscono un punto di partenza per approfondire le singole problematiche. Infatti è impensabile tentare una ad una le migliaia di vulnerabilità note sui servizi, che invece un programma può velocemente valutare e riportare.

Con questi strumenti si avrà un'indicazione di quali macchine sono più vulnerabili di altre, senza però dimenticare che il tool automatico non può fornire indicazioni precise, ma solo una linea guida su cui l'attacco si sviluppa.

12.2.2 *Enumeration*

In questa fase si effettuano connessioni dirette ai server ed interrogazioni esplicite. Queste operazioni potrebbero (a seconda della configurazione presente sui sistemi target) originare dei logs.

Attraverso l'*enumeration* si vuole giungere a identificare, sulle macchine riscontrate come raggiungibili, degli account validi, delle risorse condivise e delle applicazioni attive sulle porte in ascolto. Le tecniche utilizzate variano dai sistemi operativi delle macchine analizzate.

12.3 Attacco

12.3.1 *Gaining access*

Una volta ottenute le informazioni del punto precedente, inizia il vero e proprio attacco che ha come obiettivo quello di riuscire ad accedere al sistema remoto.

I metodi utilizzati anche in questo caso dipendono dal sistema operativo della macchina target, ma si basano sostanzialmente sulla ricerca di password corrispondenti agli utenti trovati o banali (password guessing), sullo sfruttamento di errori progettuali delle applicazioni e servizi attivi sul server (buffer overflows, attacchi data driven, ecc.) o del sistema operativo stesso.

12.3.2 *Escalating privileges*

L'obiettivo di questa fase è quello di sfruttare i risultati ottenuti nella fase precedente per ottenere il pieno controllo del sistema remoto attaccato. Si arriva a questo reperendo i file presenti sul sistema che contengono le password (/etc/passwd, SAM, ecc.) e tentando di decifrare le password in essi contenute (password cracking), oppure cercando di sfruttare gli exploits sugli applicativi presenti.

12.4 Consolidamento

12.4.1 *Pilfering*

Se si giunge a questa fase significa che si è ottenuto il pieno controllo del sistema target, quindi è bene valutare la configurazione del sistema per capire se, dove e cosa il sistema registra (logs), eventualmente si disabilita l'auditing (es. con Win NT mediante auditpol). A questo punto la macchina in oggetto può diventare un trampolino che permette di attaccare altre macchine, di conseguenza può essere utile reperire sul file system eventuali informazioni riguardanti altri sistemi

12.4.2 *Covering traces and creating backdoors*

Prima di abbandonare il sistema conquistato sono cancellati gli eventuali i logs che hanno registrato la presenza clandestina ed eventualmente installare trojan o back doors che consentano di rientrare facilmente sulla macchina in un secondo momento. Può essere utile anche installare tools nascosti quali sniffers o keyloggers al fine di catturare altre password del sistema locale o di altri sistemi ai quali ignari utenti si collegano dalla macchina violata.

Il Cliente decide a quale delle fasi sopra descritte l' *Ethical Hacking* deve fermarsi. Si giungerà al Cap. 12.3, solo se richiesto dal Cliente, in quanto le prime fasi sono sufficienti a dare un quadro completo e dettagliato delle possibili vulnerabilità.

13 Ethical Hacking Applicativo

La protezione delle reti e la cifratura dei dati sono solo una parte di quanto è possibile fare per ottenere un buon livello di sicurezza per la protezione dei dati sensibili. Spesso, infatti, le misure adottate per garantire la sicurezza dell'applicazione web che fornisce il servizio, prendono in considerazione aspetti quali il tipo di connessione, i sistemi operativi e gli strumenti (server web, mail server, ecc.) utilizzati come supporto all'applicazione, ma non entrano nel merito dell'applicazione stessa. Questo significa che se l'applicazione non è stata sviluppata da programmatori esperti in sicurezza, o non effettua tutti i necessari controlli, può essere vulnerabile e dare accesso a dati riservati, a prescindere dai componenti di sicurezza che la circondano.

Per capire se un'applicazione web è vulnerabile, e soggetta quindi ad attacchi di tipo applicativo, si rende necessaria una sua analisi approfondita ed un attacco mirato ad essa, finalizzato a mettere in evidenza tutti i punti deboli o vulnerabili che potranno in una seconda fase essere corretti.

13.1 Classi di attacco applicativo

Premettiamo che, come nel caso di hacking rivolto alle reti ed ai componenti di sicurezza, è difficile identificare *tutti* i tipi di attacco, in quanto l'approccio ad un'applicazione varia a seconda dell'applicazione stessa, ma è possibile indicare le linee guida, delle categorie di attacchi, sui quali si basa un test applicativo.

13.1.1 *Parameter tampering*

Generalmente l'accesso ai dati richiesti dall'utente remoto ad un'applicazione avviene tramite query SQL verso un database. Questo implica che può essere possibile accedere al database di back-end attraverso query SQL manipolate per ottenere dati sensibili quali ad esempio le password degli utenti, modificando in modo opportuno i parametri presenti nella pagina web.

13.1.2 *Hidden field manipulation*

Questa è una particolare forma di *parameter tampering* che utilizza i campi "hidden" presenti nelle form delle pagine web. Questi campi sono spesso utilizzati per memorizzare informazioni sulla sessione con il client remoto evitando l'utilizzo di database lato server e quindi per diminuire il grado di complessità dell'applicazione.

Purtroppo questi parametri sono assolutamente visibili all'interno del codice della pagina web quindi possono, ancora una volta, essere modificati dagli hacker che cercano di ottenere privilegi all'interno dell'applicazione.

13.1.3 *Backdoors & debug options*

In questo caso lo scopo è quello di identificare all'interno del codice della pagina web l'esistenza di flag di debug o backdoor che i programmatori spesso utilizzano durante la stesura del codice. Queste operazioni, se durante la stesura del codice sono normale pratica e diminuiscono i tempi di programmazione agevolando il lavoro degli sviluppatori, possono avere conseguenze disastrose se non sono rimosse in fase di produzione. In questa fase dell'attacco si cerca quindi la presenza di flag di debug o backdoor dimenticate dai programmatori. Vogliamo sottolineare che questa non è un'eventualità remota come può sembrare, specialmente se il rilascio dell'applicazione è avvenuto in ritardo rispetto ai tempi previsti.

13.1.4 *Cookie poisoning*

La maggior parte delle applicazioni web utilizza delle particolari tecniche che permettono di identificare l'utente remoto inviandogli un

“cookie” contenente alcune informazioni (nome utente, numero di sessione, ecc). Spesso i cookie non contengono informazioni cifrate, ed un hacker ha la possibilità di modificarli, per ingannare il programma remoto. Modificando questi valori, l’hacker potrebbe ottenere l’accesso ad utenze di altre persone o, a seconda del tipo di cookie, addirittura collegarsi ad un sito senza prima effettuare la necessaria autenticazione, impersonando quindi un altro utente.

13.1.5 *Forceful browsing*

Quando un’applicazione web non “forza” un particolare ordine nella navigazione delle pagine da parte del client remoto, un hacker ha la possibilità di richiamare direttamente alcune pagine, “saltando” le fasi di autenticazione e quindi accedendo direttamente ai dati sensibili.

13.1.6 *Cross-Site scripting*

Con questa tecnica l’hacker inserisce codice, nelle pagine web del servizio, che non proviene dal server di partenza ma da altri server. In questo modo è possibile introdurre codice errato e farlo eseguire da altri utenti ignari, in modo da poter ottenere importanti informazioni dall’utente attaccato.

13.1.7 *Buffer overflow*

La stessa tecnica utilizzata per mandare in errore un server web può essere utilizzata per mandare in errore un *programma* web, e costringerlo a compiere operazioni non previste. Questa tecnica, nel caso di applicazioni web è difficilmente sfruttabile, a meno che l’applicazione utilizzata non sia già conosciuta.

13.1.8 *Known vulnerabilities*

Anche se l’applicazione web è sicura, spesso utilizza prodotti di terze parti o librerie pubbliche o comunque note a molti, per svolgere determinate operazioni. In questo caso, se non sono state applicate patch o non sono state utilizzate le ultime versioni disponibili dai vendor, è possibile sfruttare eventuali vulnerabilità presenti nelle librerie esterne.

13.2 Tipologie di attacco

Nella seguente tabella è schematizzato l'approccio al test di *application hacking*, e gli attacchi portati verso le applicazioni.

Categoria	Nome	Obiettivo
AppDOS	Application Flooding	Assicurarsi che l'applicazione funzioni correttamente anche con l'inserimento e l'elaborazione di grosse quantità di dati.
	Application Lockout	Assicurarsi che l'applicazione non consenta ad un attacker di bloccare utenze.
Access Control	Parameter Analysis	
	Authorization	Assicurarsi che le risorse che richiedono un'autorizzazione eseguano controlli adeguati.
	Application workflow	Assicurarsi, se l'applicazione richiede all'utente di eseguire delle operazioni in una particolare sequenza, che questa sia rispettata
	Matrix Compliance	Assicurarsi che l'applicazione permetta ad un utente di accedere solamente alle risorse per cui è autorizzato.
	Implementation Consistency	Assicurarsi che il meccanismo di controllo degli accessi sia centralizzato per tutte le applicazioni e non distribuito su ogni singola applicazione.
	Completeness	Assicurarsi che tutti gli accessi all'applicazione siano verificati dal sistema di controllo degli accessi
	Identity Subversion	Verificare che l'applicazione non permetta, ad informazioni inserite dall'utente, di cambiare le autorizzazioni ottenute nella fase di autenticazione.
	Least user privilege	Verificare che l'utente abbia i privilegi minimi necessari per eseguire i propri task
	Least admin privilege	Verificare che l'amministratore abbia i privilegi minimi necessari per eseguire i propri task
	Access Mode	
Authentication	Use SSL	Assicurarsi che l'autenticazione venga richiesta utilizzando solamente il protocollo HTTPS

	Authentication bypass	Assicurarsi che il processo di autenticazione non possa essere aggirato.
Authentication.User	Credential transport over encrypted channel	Assicurarsi che la username e password siano spediti su un canale cifrato
	Username	Assicurarsi che la username non sia pubblica (indirizzo e-mail o SSN)
	Password	Assicurarsi che la password sia sufficientemente complessa.
	Password Reset	Assicurarsi che l'utente non possa resettare la propria password se non dopo aver risposto a domande con risposte predefinite (secret answer/secret question). Inoltre assicurarsi che la nuova password venga spedita tramite canali cifrati.
	Password Lockout	Assicurarsi che l'account sia bloccato per un periodo di tempo dopo che la password viene sbagliata per un numero predefinito di volte (in genere 5)
	Username enumeration	Assicurarsi che la fase di login e di recovery della password non possa essere utilizzata per enumerare gli account
	Session Token Length	Assicurarsi che il Session Token sia composto da una lunghezza adeguata per evitare il guessing.
	Session Timeout	Assicurarsi che la sessione non sia più valida dopo un determinato periodo di tempo.
	Session Reuse	Assicurarsi che il session token sia cambiato quando un utente passa da una risorsa protetta da SSL da una non protetta.
	Session Deletion	Assicurarsi che il session token venga cancellato dal lato server dopo che un utente abbia eseguito il logout dall'applicazione.
	Session Token Content	Assicurarsi che il session token sia random, non predicibile e non contenga informazioni sensibili.
Configuration. Management	HTTP Methods	Assicurarsi che il web server sia configurato per non permettere l'utilizzo di metodi che possono modificare l'applicazione (PUT, DELETE)

	Known vulnerability and patches	Assicurarsi che il web server sia aggiornato con le ultime security patch.
	Backup Files	Assicurarsi che file di backup e vecchi contenuti non siano rimasti nella document root e quindi accessibili da internet.
	Web Server Configuration	Assicurarsi che il web server sia configurato correttamente, disabilitando il directory listing e le pagine installate di default siano state rimosse.
	Web Server Components	Assicurarsi che i componenti web come le Front Page extension e i moduli apache non introducano vulnerabilità
Configuration. Management. Infrastructure	Infrastructure admin interface	Assicurarsi che le interfacce di amministrazione dell'applicazione e del web server non siano accessibili da internet.
Error Handling	Application Error messages	Assicurarsi che l'applicazione non utilizzi messaggi di errore che contengono informazioni che un attacker potrebbe sfruttare.
Data Protection	Sensitive Data in HTML	Assicurarsi che non ci siano dati sensibili nelle pagine HTML (presenti nella cache del browser)
Data Protection Transport	SSL Version	Assicurarsi che la versione di SSL utilizzata non sia affetta da vulnerabilità.
	SSL Key Exchange Methods	Assicurarsi che il web server non supporti l'anonymous key exchange
	SSL Key Lengths	Assicurarsi che il web server utilizzi una chiave di lunghezza appropriata
	Digital Certificate Validity	Assicurarsi che l'applicazione utilizzi un certificato digitale valido.
Input Validation	Script Injection	Assicurarsi l'applicazione non esegua script inseriti nelle form (CSS)
Input Validation SQL	SQL Injection	Assicurarsi che l'applicazione non esegua comandi SQL inseriti dall'utente.
Input Validation OS	OS Command Injection	Assicurarsi che l'applicazione non esegua comandi di sistema inseriti dall'utente.
Input Validation LDAP	LDAP Injection	Assicurarsi che l'applicazione non esegua comandi LDAP inseriti dall'utente

Input Validation XSS	Cross Site Scripting	Assicurarsi che l'applicazione non esegua malicious script
Output Sanitization	Output Sanitization	Assicurarsi che i caratteri speciali vengano elaborati e visualizzati correttamente
Buffer Overflow	Overflows	Assicurarsi che l'applicazione non sia soggetta ad attacchi di tipo buffer overflow
	Heap Overflow	Assicurarsi che l'applicazione non sia soggetta ad attacchi di tipo heap overflow
	Stack Overflow	Assicurarsi che l'applicazione non sia soggetta ad attacchi di tipo stack overflow
	Format String	Assicurarsi che l'applicazione non sia soggetta ad attacchi di tipo format string overflow

14 Ethical Hacking – Conclusioni

Perché è necessario effettuare anche uno scan manuale e non solo uno automatico, e quali sono le differenze tra un approccio e l'altro?

14.1 I tool automatici

Sul mercato sono presenti numerosi software in grado di effettuare un test esteso su una rete locale. Molti di questi sono facilmente reperibili su Internet, e spesso sono *Open Source*, quindi utilizzabili senza alcun costo. I tool automatici non recano danno ai sistemi su cui sono utilizzati, e costituiscono una buona protezione contro la maggior parte degli hacker detti "*script kiddies*", cioè la categoria di hacker che, non essendo esperti, utilizzano programmi (*script*, appunto) che sfruttano vulnerabilità dei servizi per tentare "a tappeto" i loro attacchi, sperando di portarne qualcuno a compimento.

I tool di scan automatico sono estremamente utili nel momento in cui si desidera controllare un gran numero di vulnerabilità note (migliaia ad oggi, ed in continuo aumento). Il processo di verifica della presenza di queste vulnerabilità si basa, ad esempio, sul riconoscimento di determinati servizi e delle loro versioni. Se si verificano particolari condizioni, quindi, è indicata la presenza di una vulnerabilità. Ad esempio, è noto che il web server di Microsoft nella versione 5.0 è affetto da un certo tipo di vulnerabilità. Se durante il test è rilevato su un server il servizio IIS ed è verificabile che la versione installata è la 5.0, è molto probabile che sia presente quella vulnerabilità. Questo meccanismo non garantisce di rilevare un problema, ma avverte che sussistono le condizioni perché quel problema si verifichi.

Appare chiaro che questo metodo può essere tratto in inganno perché possono essere dedotte vulnerabilità che di fatto non esistono. Non è detto, infatti, che se si utilizza la versione 5.0 di IIS questa non sia stata aggiornata con le ultime *patch* che correggono proprio quella vulnerabilità. In questo caso, la rilevazione effettuata è detta *falso positivo*. I falsi positivi sono il problema principale che affligge gli scan automatici.

14.2 Intrusione del Black Hat

Al contrario, se un vero hacker (il cosiddetto Black Hat) decide di portare un attacco ad un sistema, non utilizza tecniche banali che possono facilmente portare alla sua identificazione, ma fa uso soprattutto della propria esperienza per sfruttare vulnerabilità note o costruire nuovi attacchi sfruttando qualche errore di configurazione, nel software o insito nei servizi, riscontrato durante quel particolare attacco.

Ad esempio, anche se un web server non presenta CGI (programmi eseguibili lato server) con vulnerabilità note, è possibile che quelli presenti abbiano difetti di progettazione che possano portare alla scoperta di altri bug (si veda *ethical hacking applicativo*).

Abbiamo visto in questa breve descrizione quali sono le differenze tra i due approcci e perché è preferibile un utilizzo congiunto dei due strumenti. Sicuramente l'utilizzo di scan automatici permette di contenere i costi, ma non consente di ottenere il livello di profondità raggiunto con il test manuale.