]HackingTeam[| Galbusera

Galbusera

Vulnerability Assessment sulle infrastrutture di rete perimetrali

Milano

Hacking Team S.r.l.	http://www.hackingteam.it
Via della Moscova, 13 *20121 MILANO (MI) - Italy*	info@hackingteam.it
Tel. +39.02.29060603	*Fax +39.02.63118946*

STORIA DEL DOCUMENTO		
Versione	Data	Modifiche Effettuate
1.0	28 Marzo 2007	Emissione

INFORMAZIONI	
Data di Emissione	28 Marzo 2007
Versione	1.0
Tipologia Documento	Documento di progetto
Numero di Protocollo	//
Numero Pagine	17
Numero Allegati	0
Redatto da	Fabio Busatto
Approvato da	Gianluca Vadruccio

Indice

Indice 3
Executive summary 5
1 Richiesta del cliente 6
2 Metodologia 7
2.1 Analisi non invasiva 7
2.1.1 Footprinting 7
2.1.2 Scanning 7
2.2 Analisi invasiva 8
2.2.1 Enumeration 8
2.3 Attacco 8
2.3.1 Gaining access 8
2.3.2 Escalating privileges 8
2.4 Consolidamento 8
2.4.1 Pilfering 8
2.4.2 Covering traces and creating backdoors 9
3 Descrizione dell'attività svolta 10
3.1 Host 217.56.97.130 11
3.1.1 ESMTP (25/tcp) 11
3.1.2 HTTP (80/tcp) 11
3.1.3 POP3 (110/tcp) 11
3.1.4 *Unknown* (443/tcp) 11
3.2 Host 217.56.208.6 11
3.2.1 ESMTP (25/tcp) 11
3.2.2 HTTP (80/tcp) 12
3.2.3 POP3 (110/tcp) 12
3.2.4 *Unknown* (443/tcp) 12
4 Implementazione delle contromisure 13
Appendice - Security best practices 14
Principio del privilegio minimo 14
Principio della ridondanza 14

Principio della globalità 15
Principio dell'unico punto di contatto 15
Principio della modularità 15
Principio della ben definita politica di sicurezza 16
Principio della semplicità 16
Principio di Kerchhoff 16

Executive summary

Questo documento descrive l'attività di vulnerability assessment svolta da Hacking Team su alcune delle infrastrutture di rete che Galbusera utilizza per lo svolgimento delle sue attività, allo scopo di valutarne il livello di sicurezza a fronte di attacchi esterni diretti alle infrastrutture. L'intervento è limitato a due indirizzi IP esterni.

Il livello di sicurezza è risultato adeguato alla tipologia ed alla criticità dei sistemi.

I test effettuati hanno riscontrato la presenza di tutte le protezioni necessarie per impedire a potenziali intrusi di violare le politiche di utilizzo delle risorse, introducendo quindi la possibilità di effettuare attacchi più specifici in seguito alla conoscenza di dettagli implementativi non pubblici.

Un'analisi più approfondita, con un ampliamento del target dell'attività, può in ogni caso fornire nuovi elementi per valutare in maniera migliore e sicuramente più completa il livello di sicurezza effettivo dell'intera infrastruttura utilizzata.

1 Richiesta del cliente

Galbusera ha richiesto un'attività di vulnerability assessment sulle sue infrastrutture, allo scopo di verificarne il livello di sicurezza a fronte di attacchi diretti ad alcuni sistemi specifici all'interno della rete.

L'analisi deve riguardare i soli seguenti indirizzi:

- 217.56.97.130 (sede di Agrate Brianza)
- 217.56.208.6 (sede di Cosio Valtellino)

La documentazione delle attività svolte deve includere

- la descrizione dei test eseguiti
- i risultati ottenuti
- le vulnerabilità eventualmente individuate
- le relative contromisure

2 Metodologia

Il security probe condotto da Hacking Team S.r.l. è stato effettuato simulando in tutto e per tutto le attività che avrebbe compiuto un hacker. E' stato ricostruito, in altre parole, il "percorso" logico che un qualsiasi hacker percorrerebbe se volesse in qualche maniera oltrepassare le misure difensive della rete oggetto dell'analisi.

Coerentemente all'approccio metodologico di riferimento illustrato di seguito, effettuando l'attacco si è per prima cosa cercato di ricostruire la topologia e le caratteristiche dei sistemi da attaccare. In un secondo momento, tali sistemi sono stati analizzati singolarmente, e si è quindi cercato di sfruttare le debolezze presenti sugli stessi.

E' bene evidenziare che l'accesso ad una sola delle macchine da parte di un attaccante esterno, permette poi l'esposizione diretta degli altri sistemi. L'utilizzo di un "ponte" da cui accedere alle altre macchine è una pratica tipica ed efficace per ottenere il massimo da un'attività di incursione informatica.

2.1 Analisi non invasiva

2.1.1 Footprinting

Questa fase ha lo scopo di raccogliere il maggior numero di informazioni sull'obiettivo che si intende attaccare senza "toccare" 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/Ripe ed interrogazioni ai DNS.

2.1.2 Scanning

L'obiettivo dello scanning è 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 contattabili dall'esterno (Ip discovery) e, relativamente a tali ip, scoprire che servizi abbiano attivi (tcp/udp port scan) e che sistemi operativi utilizzino. Gli strumenti utilizzati sono: interrogazioni ICMP (gping, fping, ecc.), la scansione delle porte tcp e udp (strobe, netcat, nmap, amap, rascan) e fingerprint dello stack (nmap, queso).

2.2 Analisi invasiva

2.2.1 Enumeration

Con questa fase si inizia “l’analisi invasiva” infatti si effettuano connessioni dirette ai server ed interrogazioni esplicite, il che potrebbe (a seconda della configurazione presente sui sistemi target) originare dei log.

Attraverso l’enumerazione si vuole giungere a identificare, sulle macchine riscontrate come raggiungibili, degli account validi (list user accounts), delle risorse condivise (list file shares) e delle applicazioni attive sulle porte in ascolto (identify application). Le tecniche utilizzate variano dai sistemi operativi delle macchine che vogliamo analizzare.

2.3 Attacco

2.3.1 Gaining access

Una volta ottenute le informazioni del punto precedente inizia il vero e proprio attacco che ha come obiettivo il riuscire ad entrare nel 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 (password guessing), sullo sfruttamento di errori progettuali delle applicazioni e servizi attivi sul server (buffer overflow, attacchi data driven, ecc.) o del sistema operativo stesso.

2.3.2 Escalating privileges

L’obiettivo di questa fase è sfruttare i risultati ottenuti nella fase precedente per ottenere il pieno controllo del sistema remoto attaccato. Ciò si ottiene 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 utilizzando appositi exploit.

2.4 Consolidamento

2.4.1 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 stesso al fine di capire se, dove e cosa il sistema registra (log), eventualmente si disabilita l’auditing. A questo punto la macchina in oggetto può

diventare un trampolino che permetta di attaccare altre macchine, di conseguenza può essere utile reperire sul file system eventuali informazioni riguardanti altri sistemi.

2.4.2 Covering traces and creating backdoors

Prima di abbandonare il sistema conquistato vengono cancellati gli eventuali i log che hanno registrato la presenza clandestina e eventualmente installare trojan o backdoor che consentano di rientrare facilmente sulla macchina in un secondo momento. Può essere utile anche installare tool nascosti quali sniffer o keylogger al fine di catturare altre password del sistema locale o di altri sistemi ai quali ignari utenti si collegano dalla macchina compromessa.
Nel caso si riescano a superare le difese ci si fermerà comunque appena sarà possibile dimostrare l'effettiva possibilità di assumere il controllo delle macchine senza comunque apportare alcuna modifica sulle stesse.

3 Descrizione dell'attività svolta

Gli obiettivi dell'attacco sono stati i due indirizzi ip forniti dal cliente nella fase preliminare del lavoro:

- 217.56.97.130 (sede di Agrate Brianza)
- 217.56.208.6 (sede di Cosio Valtellino)

Questi indirizzi si collocano in reti assegnate a GALBUSERA DOLCIARIA S.P.A., in particolar modo ai seguenti due blocchi:

- 217.56.97.128 - 217.56.97.135
- 217.56.208.0 - 217.56.208.7

La scansione dei servizi offerti ha evidenziato i seguenti risultati:

Indirizzo IP	Servizio	Porta	Vulnerabile
217.56.97.130	ESMTP (srvk01.konsum2000.it)	25/tcp	NO
217.56.97.130	HTTP	80/tcp	NO
217.56.97.130	POP3 (srvk20.konsum2000.it)	110/tcp	NO
217.56.97.130	*Unknown*	443/tcp	NO
217.56.208.6	ESMTP (srvg01.galbusera2000.it)	25/tcp	NO
217.56.208.6	HTTP	80/tcp	NO
217.56.208.6	POP3 (srvg20.galbusera2000.it)	110/tcp	NO
217.56.208.6	*Unknown*	443/tcp	NO

3.1 Host 217.56.97.130

3.1.1 ESMTP (25/tcp)

Il servizio non risulta vulnerabile ad attacchi di tipo “buffer overflow” o ad altri attacchi simili. Inoltre non è possibile utilizzare comandi propri del protocollo per ottenere informazioni riguardanti gli utenti del sistema (enumerazione), od utilizzare il servizio per recapitare messaggi email a destinatari arbitrari, e non è quindi sfruttabile per finalità di “spam”.

3.1.2 HTTP (80/tcp)

Il servizio non risulta vulnerabile ad attacchi di tipo “buffer overflow” o ad altri attacchi simili.
Tutte le richieste vengono evase con un messaggio di errore da parte del webserver, che non lascia trasparire nessuna informazione di dettaglio sul sistema.

3.1.3 POP3 (110/tcp)

Il servizio non risulta vulnerabile ad attacchi di tipo “buffer overflow” o ad altri attacchi simili.
Inoltre non è possibile utilizzare comandi propri del protocollo per ottenere informazioni riguardanti gli utenti del sistema (enumerazione).

3.1.4 *Unknown* (443/tcp)

Il servizio chiude la connessione a seguito di qualsiasi richiesta del client, senza inviare nessun dato che permetta di identificarlo.
Dato il numero della porta utilizzata, è ipotizzabile che sia presente un servizio HTTPS, ma che sia adeguatamente protetto in maniera da permettere l’accesso solo in determinate condizioni (indirizzo di provenienza o controllo del certificato del client).

3.2 Host 217.56.208.6

3.2.1 ESMTP (25/tcp)

Il servizio non risulta vulnerabile ad attacchi di tipo “buffer overflow” o ad altri attacchi simili.
Inoltre non è possibile utilizzare comandi propri del protocollo per ottenere informazioni riguardanti gli utenti del sistema (enumerazione), od utilizzare il servizio per recapitare messaggi email a destinatari arbitrari, e non è quindi sfruttabile per finalità di “spam”.

3.2.2 HTTP (80/tcp)

Il servizio non risulta vulnerabile ad attacchi di tipo “buffer overflow” o ad altri attacchi simili.
Tutte le richieste vengono evase con un messaggio di errore da parte del webserver, che non lascia trasparire nessuna informazione di dettaglio sul sistema.

3.2.3 POP3 (110/tcp)

Il servizio non risulta vulnerabile ad attacchi di tipo “buffer overflow” o ad altri attacchi simili.
Inoltre non è possibile utilizzare comandi propri del protocollo per ottenere informazioni riguardanti gli utenti del sistema (enumerazione).

3.2.4 *Unknown* (443/tcp)

Il servizio chiude la connessione a seguito di qualsiasi richiesta del client, senza inviare nessun dato che permetta di identificarlo.
Dato il numero della porta utilizzata, è ipotizzabile che sia presente un servizio HTTPS, ma che sia adeguatamente protetto in maniera da permettere l’accesso solo in determinate condizioni (indirizzo di provenienza o controllo del certificato del client).

4 Implementazione delle contromisure

Dall'analisi condotta non appaiono vulnerabilità che possano essere sfruttate da un attaccante per ottenere un accesso non autorizzato ai sistemi presi in esame, e quindi non è necessario attuare alcuna contromisura di sicurezza in aggiunta alle politiche già esistenti.

Appendice - Security best practices

I principi metodologici che seguono sono stati impiegati nella valutazione dei livelli di sicurezza e nella formulazione delle soluzioni tecniche proposte.

Principio del privilegio minimo

"Tutto quello che non è strettamente necessario deve essere eliminato"

E' il principio più importante da seguire in materia di sicurezza. Il principio del privilegio minimo "minimum privilege" afferma che ogni soggetto all'interno di un sistema informatico (utenti, processi, programmi) deve essere in grado di accedere solamente agli oggetti del sistema (dati, accessi, flussi di dati, operazioni sui dati) di cui ha strettamente bisogno per le proprie funzioni. Il principio del privilegio minimo è fondamentale, perchè limita l'esposizione degli oggetti ad eventuali attacchi e, al tempo stesso, limita i danni subiti dall'intero sistema nel caso che un "attacco" abbia successo.

Principio della ridondanza

"Ogni meccanismo di sicurezza si può inceppare"

La sicurezza di un sistema (o di una procedura, di una funzione, di un'applicazione) non deve dipendere da un solo meccanismo di sicurezza, per quanto esso possa sembrare robusto e infallibile. E' sempre auspicabile prevedere delle soluzioni di "backup" che possano intervenire nell'evenienza di una temporanea indisponibilità di una risorsa adibita alla protezione del sistema o in presenza di un "attacco" sferrato contro la risorsa stessa.

Per esempio, è buona norma duplicare le procedure di logging quando l'auditing delle applicazioni è security-critical per il business aziendale. Oppure, assumere che le misure di sicurezza principali per il controllo dell'integrità possano in qualche modo essere "bypassate", e impiegare dei sistemi di controllo di flusso che abbiano la funzionalità di controllare che le misure di sicurezza principali siano ben funzionanti.

A supporto di quanto è stato detto, bisogna osservare che tutte le tecnologie di security soffrono di un'obsolescenza assai più rapida rispetto agli strumenti software convenzionali.

La qualità e l'efficacia degli attacchi che possono essere effettuati contro un sistema informatico è in costante evoluzione, e per questa ragione è necessario che le misure di sicurezza rispecchino le nuove tecniche di attacco non appena queste diventano note.

Internet è un formidabile catalizzatore del processo evolutivo “nuovo attacco - nuova misura di sicurezza per rendere inefficace l’attacco - nuovo attacco in grado di neutralizzare la misura di sicurezza precedente”. È opportuno ipotizzare che anche il personale interno all’azienda possa essere in grado di procurarsi informazioni e tecnologie sufficienti a sfruttare le debolezze della infrastruttura.

Principio della globalità

“Una catena è forte quanto il suo più debole anello”

Un’infrastruttura informatica complessa è composta da numerosi elementi strettamente interconnessi.

La sicurezza dell’intera infrastruttura è il risultato della sicurezza dei singoli elementi e, soprattutto, della sinergia che i singoli elementi, una volta raggruppati, riescono a formare. Non ha senso rafforzare massicciamente la sicurezza di un solo elemento lasciandone vulnerabile un altro: in tal caso, chi compie la frode informatica sfrutterà l’insicurezza di quest’ultimo per violare la sicurezza dell’intero sistema.

Chi è intenzionato a violare la sicurezza del sistema cercherà di “passare” per la strada più breve, cioè per quella con il più conveniente rapporto costi / benefici. Spesso la via più facile per accedere illegalmente alle informazioni non è affatto tecnica.

Talvolta è preferibile, per l’hacker, acquisire le informazioni che desidera corrompendo un addetto interno piuttosto che tentando un attacco tecnico ad alta sofisticazione come la crittoanalisi di un algoritmo crittografico con cui sono protetti i dati.

Principio dell’unico punto di contatto

“E’ più facile controllare un unico punto di passaggio”

E’ buona norma concentrare le funzioni di sicurezza applicative, di rete, ecc. su di un numero esiguo di sistemi, in maniera che la sicurezza dell’intera infrastruttura dipenda da pochi punti altamente controllabili.

Principio della modularità

“E’ più facile controllare la sicurezza di piccoli oggetti”

Oggetti piccoli sono più facilmente gestibili e controllabili. Nel caso che un oggetto fallisca, la sicurezza dell’intera infrastruttura può essere preservata. Un oggetto piccolo, inoltre, ha una

complessità inferiore rispetto ad un oggetto grande e integrato ed è quindi più difficile che al suo interno siano contenute debolezze applicative ("bugs"). Questo principio permette anche di individuare con maggiore facilità le parti più critiche del sistema, dando la possibilità di interventi il più possibile mirati nell'evenienza di aggiunte, potenziamenti o aggiornamenti di ciascuna delle componenti.

Principio della ben definita politica di sicurezza

"Nel dubbio, meglio negare che permettere"

Nella progettazione di un sistema di sicurezza sono possibili due approcci:

- Quello che non è espressamente permesso è proibito;
- Quello che non è espressamente proibito è permesso.

In linea generale, il primo approccio è sempre preferibile dal punto di vista della sicurezza.

Principio della semplicità

"KISS: Keep It Simple Stupid"

La semplicità va d'accordo con la sicurezza. Ma complessità va d'accordo con la mancanza di visibilità da cui, immancabilmente, scaturisce l'insicurezza. Le componenti di un sistema di sicurezza devono essere il più semplici possibili, affinchè il sistema risulti facile da usare e da gestire. E' un errore storico quello di pensare che un sistema grande e complesso debba essere sicuro. Un sistema grande e complesso è tipicamente difficile da analizzare, fino a diventare oscuro.

Quello che per noi è difficile da capire può apparire cristallino agli occhi di chi vuole compiere una frode informatica. La semplicità, quindi, gioca dalla nostra parte: più un oggetto è semplice, più una procedura è comprensibile e maggiori sono le probabilità che sia sicura. E' noto dall'ingegneria del software che i programmi complessi hanno più "bugs" e tra questi è probabile che ce ne siano alcuni relativi alla sicurezza .

Principio di Kerchhoff

"Chi compie la frode conosce sempre tutti i dettagli implementativi"

Se la robustezza di un sistema di sicurezza è basata sul fatto che non siano pubblicamente noti gli "internals", gli algoritmi o le specifiche tecnologie usate, allora il sistema in questione è assai insicuro. E' un approccio errato credere che, al fine di aumentare la sicurezza, sia meglio

mantenere la propria tecnologia di difesa segreta piuttosto che lasciare che tale tecnologia venga visionata da un grande numero di esperti. Assumere che sia un compito difficile effettuare il “reverse engineering” di un’applicazione è un grave errore, un errore che purtroppo viene commesso da molti. I migliori oggetti di sicurezza sono quelli che impiegano algoritmi e protocolli pubblici che sono stati attaccati, analizzati e corretti per anni dai migliori esperti di sicurezza. E’ storicamente noto come moltissimi prodotti definiti proprietari sono risultati del tutto insicuri ed inadeguati una volta che i loro internals sono stati scoperti e resi pubblicamente noti.