]HackingTeam[| b@nca 24-7

Banca 24-7

Ethical Hacking

Milano

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

TORIA DEL DOCUMENTO		
Versione	Data	Modifiche Effettuate
1.0	02 Agosto 2005	Prima stesura
1.1	05 Agosto 2005	Definizione soluzioni ed emissione
//	//	//
//	//	//

INFORMAZIONI	
Data di Emissione	05 Agosto 2005
Versione	1.1
Tipologia Documento	Documento di Progetto
Numero di Protocollo	//
Numero Pagine	29
Numero Allegati	0
Redatto da	Massimo Chiodini Marco Valleri
Approvato da	Gianluca Vadruccio

INDICE

1 Architettura e descrizione del progetto

1.1 Ambiente di lavoro

L'analisi dei sistemi di perimetro di *Banca 24-7* sono state effettuate da *HackingTeam srl* presso la propria sede di Milano, simulando un eventuale attacco da parte di un *hacker* su internet. L'analisi dei sistemi e delle applicazioni ~~internet~~ e' stata invece svolta presso la rete del Cliente, utilizzando una normale postazione di lavoro con accesso alla rete locale.

1.2 Target

I sistemi *target,* definiti in fase di pianificazione e concordati con il cliente (per l'attività di analisi perimetrale) sono:

- 64.94.82.1
- 64.94.82.13
- 64.94.82.14
- 62.94.244.60

L'analisi interna è stata invece svolta con un approccio *Black Box* (nessuna informazione sulla rete o sui sistemi critici). Successivamente, il Cliente ha indicato i seguenti sistemi come critici, e l'attività si è focalizzata su di essi:

- 10.16.29.70
- 10.16.29.120
- 10.208.17.21
- 10.208.17.24

1.3 Attività svolte

L'attività di *assessment* segue una metodologia ben consolidata che prevede il reperimento del maggior numero di informazioni utili per poter, in seguito, portare con successo un attacco verso i sistemi *target*.

In generale l'attacco ad un sistema sfrutta vulnerabilità intrinseche nei servizi (sia di natura logico-architetturale, sia di natura implementativa) per indurre comportamenti anomali in quest'ultimi, le cui conseguenze possono essere le più disparate: crash dell'applicazione, accesso ai sistemi su cui i servizi sono in esecuzione, ecc.

Allo scopo di inquadrare il tema della sicurezza, sia in termini di "opportunità" offerte all'intrusore, sia di minacce per le potenziali "vittime", si dà una sintetica descrizione delle fasi che compongono un attacco.

I concetti e la terminologia introdotti saranno utilizzati nel presente documento per descrivere i risultati dell'*assessment* svolto.

Le attività effettuate sul perimetro e sulla rete interna di si possono riassumere in:

- *Network analysis*: comprende tutte le attività di *reverse engineering* delle rete del cliente, che va' dalla raccolta di informazioni di pubblico dominio come i nomi e gli indirizzi assegnati, fino all'analisi dei componenti di connettività ed instradamento verso internet.
- *Fingerprinting* attivo e passivo dei sistemi: il *fingerprinting* consiste nell'individuazione dei sistemi attivi e della loro catalogazione in base alle risposte a sollecitazioni non invasive sui protocolli abilitati.
- *Scanning*: e' la fase che conclude l'attività non invasiva, che consente di rafforzare le ipotesi fatte durante il l'attività di *fingerprinting* e di rilevare servizi e applicazioni attive sui sistemi.
- *Enumeration*: lo scopo è quello di enumerare le risorse dei sistemi in termini di servizi aperti al pubblico e di raccogliere informazioni quanto più dettagliate sulla tipologia e versione di quest'ultimi, allo scopo di rintracciare vulnerabilità che affliggono le versioni dei software utilizzati.
- *Attacco*: e' la fase più complessa e delicata dell'intera attività, in cui tutte le informazioni precedentemente raccolte vengono validate e utilizzate con l'obiettivo di compromettere i

sistemi *target*. Le modalità e le tecniche che vengono utilizzate possono variare notevolmente a seconda dello scenario.

- *Privileges Escalation*: l'attività consiste nel tentativo di elevare i privilegi con cui si accede ad un sistema compromesso durante la fase precedente, allo scopo di consentire l'accesso al maggior numero di risorse possibile (documenti riservati, servizi, applicazioni, ecc.)

Le fasi sono state effettuate utilizzando una serie di *tools* proprietari e/o di pubblico dominio, come ad esempio strumenti di analisi dei protocolli, *port scanner*, *Sniffer*, *remote exploit*, ecc.

2 Analisi perimetrale - risultati ottenuti

2.1 Analisi dei target

In questo paragrafo sono elencati i risultati ottenuti in seguito all'analisi della topologia della rete e delle macchine e dei servizi in essa contenuti.

2.1.1 Analisi rete target

Di seguito viene riportato il disegno di rete relativo all'infrastruttura del Cliente ipotizzato da HT successivamente alle attività di *footprinting, scanning e network reconnaissance.* Questo grafico è esemplificativo di come un attaccante esterno, che non possieda alcuna informazione riguardo alla topologia in esame, può immaginare la rete del Cliente[1].

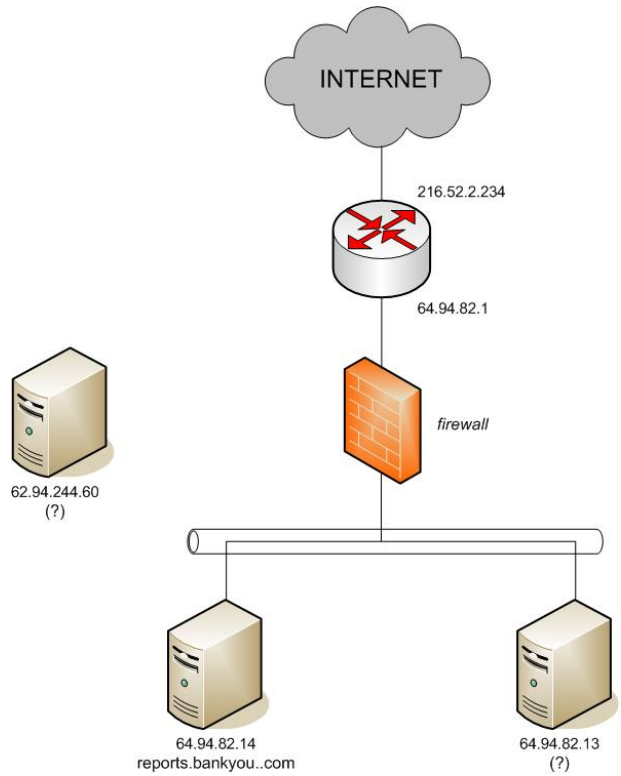


Figura 1 – Topologia ipotizzata della rete in esame

[1] Nella figura sono riportate solo le macchine corrispondenti agli indirizzi IP su cui il Cliente ha richiesto l'analisi.

2.1.2 Analisi macchine target

Di seguito viene riportata una descrizione della macchine analizzate basata sui dati ottenuti durante lo svolgimento delle attività di *ethical hacking*. Insieme alle informazioni di carattere generale (servizi aperti, *OS fingerprint*, etc.), vengono riportate le eventuali vulnerabilità puntuali riscontrate e il relativo livello di criticità.

2.1.2.1 64.94.82.1

General Info		
OS fingerprint	Cisco IOS 11.X/12.X	
Open TCP services	Number	Service
	23	telnet
Open UDP services	Number	Service
	123	NTP
	161	SNMP

- Questo indirizzo IP risulterebbe essere relativo all'interfaccia interna del *router* che gestisce la connettività internet della rete del Cliente.
- Il servizio *telnet* esposto richiede l'autenticazione tramite *username* e *password*. Un attacco di tipo *brute-force* non ha prodotto risultati degni di nota. Si consiglia tuttavia di non utilizzare questo servizio per l'amministrazione del router (se questa viene compiuta dal Cliente), in quanto il protocollo prevede l'invio delle credenziali sulla rete in *clear-text*.
- Il servizio SNMP non ha risposto a richieste effettuate utilizzando i *community-name* più comuni.
- La macchina risponde a pacchetti di tipo *ICMP Echo Request* (Ping).

2.1.2.2 64.94.82.13

- La macchina non risulta avere servizi pubblici utilizzabili dalla rete da cui lo *scan* è stato effettuato (non è possibile confermarne la reale presenza).

2.1.2.3 64.94.82.14 – reports.bankyou.com[2]

General Info		
OS fingerprint	Windows 2000	
Open TCP services	Number	Service
	443	HTTPS

- La macchina risulta protetta da un dispositivo di filtraggio del traffico (*firewall*).
- La macchina ospita il portale web d'accesso al sistema di *Reports.*
- Sebbene la macchina non presenti vulnerabilità di tipo sistemistico, l'applicazione web ospitata risulta essere vulnerabile ad attacchi di tipo SQL-Injection (si veda paragrafo 2.2).

2.1.2.4 62.94.244.60

- La macchina non risulta avere servizi pubblici utilizzabili dalla rete da cui lo *scan* è stato effettuato (non è possibile confermarne la reale presenza).

2.2 Analisi applicativa reports.bankyou.com

Hacking Team ha svolto un attacco di tipo applicativo con approccio *Black Box* (nessuna informazione fornita dal Cliente) al portale reports.bankyou.com. L'accesso a tale portale risulta protetto da un *form* di autenticazione che richiede l'inserimento della coppia di credenziali *username/password*. La gestione lato-server dei dati immessi dall'utente, all'interno di questo form, e' risultata priva degli adeguati *sanity check*, e questo ha permesso di effettuare un attacco di tipo SQL Injection verso il sistema di autenticazione.

Il codice JSP sul server, che gestisce il sistema di *login*, effettua delle *query* SQL sul database di *backend* (Oracle) per ottenere i dati necessari all'autenticazione e alla profilatura degli utenti. Con

[2] Il servizio HTTPS sulla macchina 64.94.82.14 era risultato filtrato in seguito ad una prima scansione delle porte TCP. Questo lascia supporre che la macchina in questione (e plausibilmente anche le macchine corrispondenti agli indirizzi 64.94.82.13 e 62.94.244.60) sia protetta da un sistema di *blacklist* sul *firewall* per prevenire i *port scan*, oppure che il servizio sia risultato temporaneamente inattivo. In entrambi i casi è possibile supporre che su questa, e sulle altre macchine che ricadono nella stessa casistica, siano attivi dei servizi che non sono stati rilevati dall'attività di *scanning*. Nel caso sia presente un sistema anti-scan potrebbe comunque essere possibile effettuare una scansione completa dei servizi con un adeguato *effort* temporale.

la tecnica della SQL Injection, un attaccante e' in grado di modificare le *query* SQL effettuate dal codice JSP, al fine di aggirare il sistema di autenticazione o di accedere al contenuto informativo del database.
Qui di seguito viene riportata una lista contenente le principali problematiche che hanno permesso di portare con successo un attacco di questo tipo, o che ne hanno accentuato l'impatto:

- **Utilizzo diretto di dati forniti dall'utente all'interno di query SQL**: se non vengono effettuati opportuni *sanity check,* o un corretto *escaping* dei dati, l'utente può utilizzare alcuni caratteri "speciali" che non verranno correttamente interpretati dal parser SQL, modificando il significato originale delle query. Esempi di questi caratteri sono "'" (apice), ";" (punto e virgola), etc.
- **Mancata gestione degli errori generati dal backend**: Se i messaggi di errore generati dal parser SQL sul database di *backend* vengono inviati all'utente (e non sono gestiti lato-server), un attaccante può utilizzare i dati in essi contenuti per ottenere preziose informazioni sulla struttura interna del database.
- **Mancata profilatura utenti database**: Se viene utilizzato un unico utente per far accedere le applicazioni al database, la compromissione di una singola *query* SQL permette all'attaccante di accedere a tutto il contenuto informativo del DB (anche ai dati che non sono legati all'applicazione compromessa).

Una volta avuto accesso all'applicazione, si è potuto constatare come anche i *form* interni dell'applicazione (es: *form* di ricerca) soffrano della medesima vulnerabilità, e permettono di portare attacchi simili a quello precedentemente descritto.

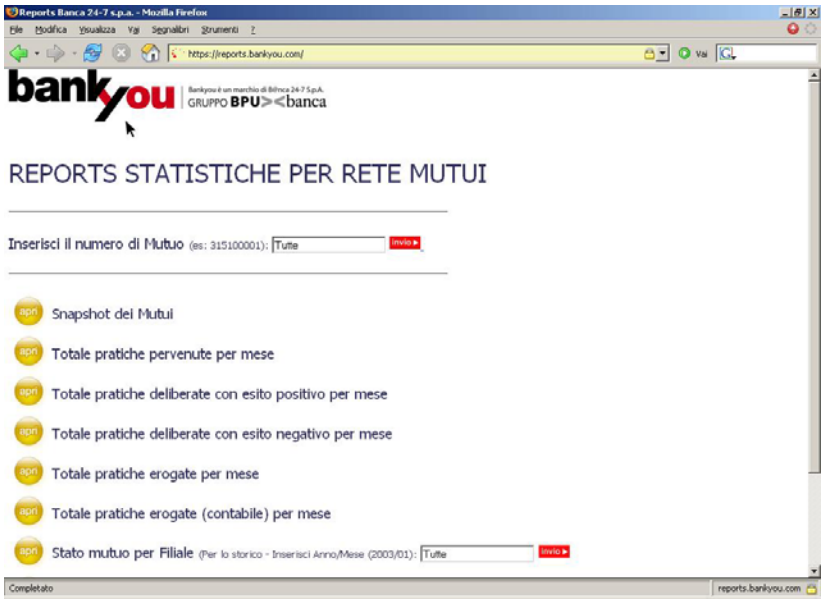


Figura 2 – Accesso all'applicazione senza credenziali

Di seguito viene riportato un piccolo estratto dei dati a cui si è potuto avere accesso in seguito alla compromissione dell'applicazione:

- CARTE DI CREDITO RICARICABILI:

ABDELLAH AZEQQAD 01-DEC-07 4295260902181607 624
ABDULLAH ATTAKROURY 01-DEC-07 4295260903733133 1577
ADELINA ATZENI 01-DEC-07 4295260903730238 1287

- MOVIMENTI:

DATA_ACQUISTO: 2003-04-05
DESCRIZIONE: GAEC FERME SCHRU17206 68ORBEY
IMPORTO: 681
NUMERO_CARTA: 4935120036896239

- PERIZIE:

Oggetto:Appartamento poso al piano primo composto da corridoio d`ingresso e disimpegno, soggiorno, cucina abitabile, due camere e servizio.`
NomePerito:Quai Angelo
Comune:GHISALBA
Mutuo:43000

- MUTUI DELIBERATI PER CI 87:

Mutuo:	NDG Nome Cognome:	Importo:	Data Deliberato:	Declinata:
005100001	441596839_PALMIERI_PIETRO	120000	2003-12-17	SI
005100002	174528687_ROZZI_ERNESTO	62000	2003-11-11	NO

Per maggiori dettagli riguardo alla tecnica SQL injection e alle possibili contromisure, **fare riferimento al capitolo 4 (contromisure).**

N.B. Il Cliente ha già effettuato degli opportuni interventi correttivi per far fronte alla problematica rilevata.

3 Analisi intranet - risultati ottenuti

In questo capitolo vengono riportati i risultati dell'analisi svolta nella *intranet* del Cliente. Nel paragrafo 3.1 viene presentata un'analisi completa dei servizi e delle relative vulnerabilità potenziali (verificate dove possibile) per i server identificati come critici dal Cliente. In seguito sono descritte le varie "aree" che sono state oggetto di analisi, secondo un approccio *Black Box,* e sono riportati tutti i sistemi e le applicazioni a cui si e' potuto avere accesso, o che si sono potuti compromettere durante lo svolgimento dell'attività, ognuno accompagnato dalle cause che hanno reso possibili tali accessi non autorizzati.

3.1 Server rilevanti

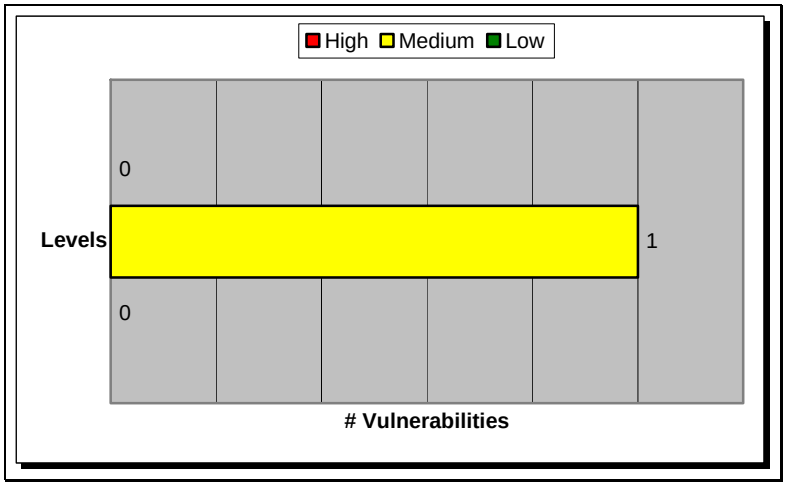
Di seguito vengono riassunti i risultati delle analisi svolte:

- sui server indicati dal cliente come strategicamente importanti.
- sui server dove sono state riscontrate vulnerabilità degne di nota.

Insieme alle informazioni di carattere generale vengono riportate, per ogni macchina, le vulnerabilità *sistemistiche* riscontrate (dove presenti) e le relative proposte per il *fixing*.

N.B. Trattandosi di macchine in produzione, non sono state testate le vulnerabilità che avrebbero potuto compromettere il buon funzionamento dei sistemi, e non sono stati portati attacchi di tipo *Denial Of Service* (negazione del servizio).

3.1.1 10.16.29.70 - *server8.bankyou.it*

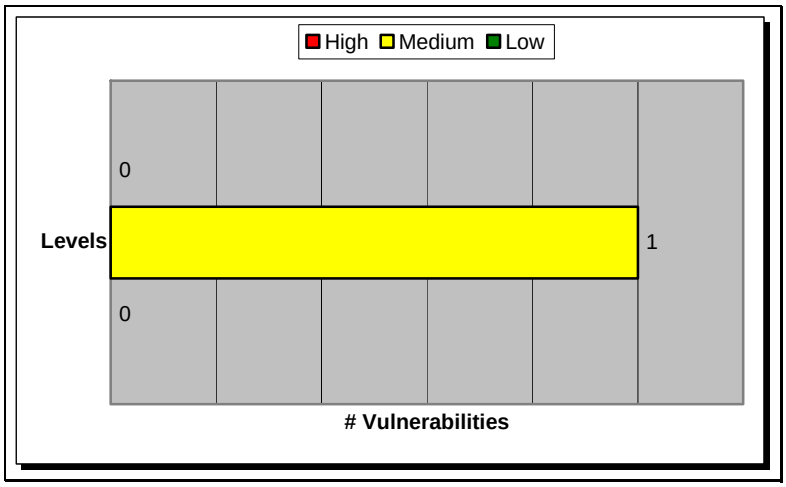


General Info		
OS fingerprint	Windows 2003	
Open TCP services	Number	Service
	53	dns
	88	kerberos
	135	ms-rpc
	139	netbios-ssn
	389	ldap
	445	ms-ds
	464	kpasswd5
	593	http-rpc-epmap
	636	ldap ssl
	1025	drs
	1027	drs
	1040	NtFrs
	3268	-
	3269	-
	5800	vnc http
	5900	vnc
Open UDP services	Number	Service
	53	dns
	67	bootps

- La macchina risulta essere il *Domain Controller* per il dominio Microsoft “BANKYOU” (per maggiori informazioni riguardanti il dominio BANKYOU si veda il paragrafo 3.2).

Vulnerabilities					
#n	Level	Name	Description	Threat	Fix
M1	Medium	Microsoft NULL Sessions [Port :445/TCP]	E’ possibile ottenere la lista degli utenti, dei gruppi, delle policy, etc. senza dover fornire credenziali valide.	Sebbene non rappresenti di per se una vulnerabilità’, la possibilita’ di ottenere queste informazioni puo’ aiutare enormemente un successivo attacco *PasswordGuessing* o *BruteForce.*	Consultare le *Best Practice* di Microsoft al link: http://support.microsoft.com/default.aspx?scid=http://support.microsoft.co:80/support/kb/articles/Q246/2/61.ASP&NoWebContent=1

3.1.2 10.16.29.120 – *nodo1.bankyou.it*



General Info		
OS fingerprint	Windows 2000	
Open TCP services	Number	Service
	25	smtp
	26	-
	80	http

	110	Pop3
	111	rpcbind
	135	Ms-rpc
	139	Netbios-ssn
	443	https
	445	Ms-ds
	593	http-rpc-epmap
	691	resvc
	993	imaps
	995	Pop3s
	3372	msdtc
	3389	Term Serv
	5800	vnc http
	5900	vnc
	6050	arcserve
	8081	-
Open UDP services	Number	Service
	111	portmap
	137	Netbios-ns

- La macchina risulta essere il nodo attivo di un *cluster* di Microsoft Exchange Server.
- Per mezzo della vulnerabilità M1 è stato rilevato un utente di sistema (con elevati privilegi) avente una *password* uguale allo *username*. **Per maggiori dettagli sull'effettivo impatto di questa problematica fare riferimento al paragrafo 3.5.2**.
- E' possibile accedere allo *share* di rete EXCHANGESRV.LOG senza fornire credenziali. In questo *share* sono presumibilmente contenuti i log relativi al servizio Microsoft Exchange. Di seguito viene riportato un piccolo estratto di tali log:

```
# Message Tracking Log File
# Exchange System Attendant Version 6.0.6249.0
# Date       Time     client-ip     Client-hostname   Partner-Name       Server-hostname
2005-7-17 0:1:6 GMT  10.16.26.254  fw-1              -                  EXCHANGESRV

server-IP    Recipient-Address     Event-ID
10.16.29.123 supporto@banca247.it  1019
```

Vulnerabilities					
#n	Level	Name	Description	Threat	Fix
M1	Medium	Microsoft NULL Sessions [Port :445/TCP]	E' possibile ottenere la lista degli utenti, dei gruppi, delle policy, etc. senza dover fornire credenziali valide.	Sebbene non rappresenti di per se una vulnerabilità', la possibilita' di ottenere queste informazioni puo' aiutare enormemente un successivo attacco *PasswordGuessing* o *BruteForce.*	Consultare le *Best Practice* di Microsoft al link: http://support.microsoft.com/default.aspx?scid=http://support.microsoft.co:80/support/kb/articles/Q246/2/61.ASP &NoWebContent=1
M2	Medium/Low	BrightStor ARCServe UniversalAgent Buffer Overflow[3] [Port: 6050/TCP]	Inviando un particolare pacchetto malformato e' possibile eseguire un *buffer overflow* nella memoria del servizio.	Un attaccante, da remoto, potrebbe eseguire codice arbitrario sulla macchina o causare un arresto del servizio (D.o.S.)	Applicare gli aggiornamenti indicati dal *vendor* per la versione in uso del software: http://www.securityfocus.com/bid/13102/solution

[3] I tentativi di *exploiting* di questo servizio non hanno prodotto risultati di rilievo. Per questo motivo non è stato possibile determinare con esattezza l'effettiva presenza di questa vulnerabilità, ed il suo effettivo livello di rischio è stato abbassato a Medium/Low. Si consiglia di verificare manualmente la presenza degli aggiornamenti indicati nei *Fix*.

3.1.3 10.208.17.21 - *infra247.bankyou.it*

General Info		
OS fingerprint	SunOS 5.9	
Open TCP services	Number	Service
	7	echo
	9	discard
	13	daytime
	19	chargen
	21	ftp
	22	ssh
	23	telnet
	37	time
	79	finger
	111	rpcbind
	512	exec
	513	login
	514	shell
	515	printer
	540	uucp
	898	sun mgmt
	1521	oracle
	4045	lockd
	6003	-
	6112	dtspc
	7100	font
	7777	http
	8080	http-proxy
	9090	zeus-admin

- La macchina sembra gestire il sistema di *Single Sign On* (Oracle-based) per l'accesso alle applicazioni intranet.
- E' stata riscontrata la presenza di un utente di sistema creato di default all'installazione di Oracle. Con le credenziali oracle/oracle è possibile accedere alla macchina tramite ad esempio il servizio telnet. Per maggiori dettagli sull'impatto di questa problematica fare riferimento ai paragrafi 3.5 e 3.6.
- La macchina pubblica numerosi servizi inutilizzati. Si consiglia di eliminare tutti i servizi non necessari al corretto funzionamento del sistema (es: *echo*, *discard*, etc.).
- La macchina pubblica vari servizi di amministrazione remota con protocollo *clear-text* (es: telnet). Si consiglia l'utilizzo dei soli protocolli cifrati (es: ssh).

3.1.4 10.208.17.24 - *appl2.bankyou.it*

General Info		
OS fingerprint	SunOS	
Open TCP services	Number	Service
	80	http

- La macchina ospita il portale di accesso web alle applicazioni *intranet*. L'accesso a tali applicazioni viene gestito tramite un sistema di SSO oracle-based.
- Fra le componenti web accessibili è stata rilevata la presenza di */fcgi-bin/echo.* Sebbene di per se non rappresenti un problema per la sicurezza, se ne consiglia la rimozione.
- Per l'analisi delle applicazioni ospitate su questo server fare riferimento al paragrafo 3.7.

3.2 Dominio BANKYOU

Il dominio *BANKYOU* è controllato dal server [*server8* – 10.16.29.70] che utilizza il sistema operativo Microsoft Windows 2003. I risultati delle attività su questa macchina non hanno messo in evidenza vulnerabilità rilevanti a livello di servizi, e i tentativi di attacco portati non hanno sortito alcun effetto.

L'unica cosa da segnalare è che il sistema permette di enumerare risorse quali utenti, gruppi e *policy* di autenticazione, tramite la tecnica delle *null session*.

Questa tecnica consiste nell'accedere ai servizi *rpc* (tramite il servizio microsoft-ds [porta *TCP 445*] o netbios-ssn [porta *TCP 139*]) specificando *username* e *password* nulli, ed effettuare chiamate *rpc* allo scopo di recuperare informazioni sul sistema.

Utilizzando un semplice *tool* è possibile recuperare la lista degli utenti del dominio o le password policy: informazioni utili per tentare attacchi di tipo "*password guessing*". Di seguito viene presentato un piccolo estratto dalla lista degli utenti di dominio.

```
administrator
aerrico
aferaco
agarro
amaccarrone
amerlini
[…]
```

In seguito al reperimento di queste informazioni, è stato possibile rilevare come il server utilizzi un meccanismo di *lockout* degli account in seguito a un numero eccessivo di tentativi di accesso falliti (5); tale configurazione diminuisce le possibilità di un attacco *password guessing.*
Tale tipo di attacco può avere un impatto molto alto sull'intera rete. E' sufficiente infatti un solo account con alti privilegi (del dominio o del PDC) che abbia una password debole, per poter aver accesso alle credenziali di tutti gli utenti del dominio e, di conseguenza, a tutte le sue risorse.
Sono stati effettuati dei test in tal senso su *una parte* delle utenze ottenute in precedenza, ma senza ottenere risultati di rilievo (anche per via delle restrittive password policy imposte dalle configurazioni del dominio).

E' comunque possibile inibire il reperimento di tali informazioni sul dominio (utenze, gruppi, policy, etc) creando nel registry, sotto la *hive* "*HKLM\SYSTEM\CurrentControlSet\Control\LSA*", l'entry *RestrictAnonymous* di tipo REG-DWORD settata al valore "1".

Infine, dato che il sistema di SSO utilizza le medesime credenziali del dominio, è importante notare come una sua eventuale compromissione porterebbe anche alla compromissione del dominio Microsoft e di tutte le risorse da esso gestite.

3.3 Apparati di rete

Per la parte di infrastruttura di rete sono stati analizzati due *switch* Cisco Catalyst (SW1 [10.16.29.251] e SW2 [10.16.29.253]) assestati nel medesimo segmento di rete da cui e' stato effettuato l'attacco.
Nello specifico, l'interfaccia di amministrazione WEB su SW2 risulta passibile di una vulnerabilità che permette di avere **accesso all'apparato con privilegi di amministrazione**. Le password di accesso, reperite in questo modo nelle configurazioni del sistema, sono risultate valide anche per l'accesso a SW1.
Per maggiori dettagli fare riferimento alle soluzioni proposte dal vendor all'URL:
`http://www.cisco.com/warp/public/707/IOS-httplevel-pub.html`

3.4 Analisi traffico di rete

In seguito ai test effettuati è stato rilevato come la rete non presenti particolari contromisure contro l'intercettazione del traffico da parte di un attaccante (*sniffing, arp poisoning, dhcp spoofing,* etc.). Unitamente al fatto che molti dei protocolli utilizzati nella rete (sistemi di amministrazione remota, connettività verso i database, applicazioni web, etc.) sono in chiaro, questo potrebbe compromettere la sicurezza dei sistemi e delle applicazioni ospitate.

Di seguito un piccolo estratto dei dati catturati dalla rete durante l'attività:

```
cognome TERRACINA
nome ELISABETT
NDG 262564919
Codice Fiscale TRRLBT64D69H501M
Nome azienda MATHERIA S.R.L.
Partita IVA azienda 06239981001
Residenza - Indirizzo VIA DI VILLA EMILIANI 21
Residenza - Cap 00197
Residenza - Provincia RM
Prefisso Telefono 0006
Numero Telefono 05817566
Prefisso Cellulare
Numero Cellulare
Nascita - Luogo ROMA
Nascita - Provincia RM
Nascita - Data 29 Apr 1964
Nazionalità ITA
Corrispondenza - Indirizzo VIA MAROSTICA 44


PAN 4CF23BE01BC1C44C
Tipo carta principale carte aggiunte
PIN NO PIN
EMBOSSING NO EMBOSSING
Prodotto VKC-1-6-10-0006-0576-5048
Fido 160000
Data Richiesta Carta 05 Dec 2002
  NO ECREDIT
Data Invio Richiesta SECETI Campo da aggiungere
Stato FP - FRAUD PICKUP
  Variazioni

richiesta carta id 920020054140
NDG titolare 262564919
data creazione 2004-06-29
```

```
flag_multifunzione_f 0
CAB addebito 03208
loyalty programs 0
iniziativa_f 0006
data_scadenza_punti_f
data_abilitazione_conto_f 29 Jun 2004

user: rgualandris  password:rgualandris  INFO: http://appl2/CarteTraker/pri/menu.do
```

Per un'analisi delle possibili contromisure fare riferimento al capitolo 4.

3.5 Password/Profile policy

L'analisi della metodologia di profilatura degli utenti, e delle policy per l'assegnazione e il cambio delle credenziali, sul dominio BANKYOU non ha evidenziato vulnerabilità degne di nota, ed il suo livello di sicurezza e' risultato adeguato alla sensibilità dei dati trattati e dei servizi coinvolti (vedere paragrafo 3.2).

Viceversa, l'attività di test ha potuto evidenziare come molte delle applicazioni installate sui server utilizzino delle utenze (interne o di sistema) di default. Di seguito alcuni esempi:

- **Utenza oracle/oracle**: la quasi totalità delle macchine analizzate, su cui è installato un server Oracle, risulta avere un utenza *oracle* con password *oracle,* e permette l'accesso via telnet. Questo utente di sistema ha, mediamente, dei privilegi elevati nei confronti di tutte le risorse che hanno a che fare con il database server. Alcuni esempi:
  - 10.16.29.118
  - 10.16.29.10[4]
  - 10.208.17.21

- **Utente SA (MS-SQL)**: alcune delle macchine analizzate, sulle quali è installato Microsoft SQL Server, hanno un utenza di default per l'amministrazione del database (utente:SA password:[blank]). Tramite questa utenza è possibile non solo accedere al contenuto informativo del database, ma anche, utilizzando le *extended stored procedure* (es: xp_cmdshell), ottenere una *shell* interattiva sulla macchina con privilegi amministrativi.

[4] Questa macchina presenta anche una vulnerabilità del sistema di login che permette di **accedere al sistema come utente *root*** senza dover fornire credenziali valide. Installare la *patch* fornita dal *vendor* Sun Patch 110668-03.

Ad esempio:

```
C:\WINDOWS\system32\httmp>ipconfig /all
ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : crif1050
   Primary Dns Suffix  . . . . . . . : bankyou.it
   Node Type . . . . . . . . . . . . : Unknown
   IP Routing Enabled. . . . . . . . : No
   […]
```

Una volta avuto accesso ad una macchina, con i metodi sopra descritti, è possibile analizzarne il contenuto informativo alla ricerca di ulteriori credenziali, o *trust-relationships,* che permettano di avere accesso ad altri sistemi. Gli esempi più comuni possono essere il reperimento dei file */etc/passwd* e */etc/shadow* (sistemi Unix) contenenti le credenziali degli utenti di sistema, file di tipo *rhosts* che definiscono le relazioni di *trust* per il servizio rlogin, relazioni di *domain trusting* (sistemi Windows), file contenenti le *history* delle shell contenenti gli ultimi comandi eseguiti dall'utente legittimo, *script* contenenti credenziali cablate al loro interno.

L'attività di analisi ha messo in luce come all'interno della rete del Cliente viene fatto largo uso dei sistemi appena descritti . Si consiglia pertanto di ridurre al minimo le relazioni di *trust* e le credenziali cablate all'interno degli *script* di automazione.
Un classico esempio di questo tipo di problematica è una macchina di sviluppo dove sono contenuti gli script per effettuare il *deployment* automatico sui server in produzione: l'eventuale compromissione di tale macchina porterà alla compromissione del contenuto informativo del sistema di produzione.

Di seguito vengono riportati due esempi significativi delle macchine a cui si è potuto avere accesso con i metodi sopra descritti.

3.5.1 infra247.bankyou.it

Accedendo alla macchina tramite il servizio *telnet* (utente oracle/oracle) è possibile visualizzare le configurazioni per la connettività del server Apache verso il database Oracle. Tramite queste configurazioni è possibile ottenere le credenziali di accesso dell'utente Oracle utilizzato dal sistema per la gestione del servizio di SSO (ORASSO). In questo modo dovrebbe essere possibile ottenere la lista completa delle credenziali degli utenti del sistema di SSO. Queste credenziali dovrebbero

risultare valide anche per l'accesso alle risorse del dominio Microsoft Windows (essendone una replica). Tale possibilità non è stata comunque analizzata a fondo, in quanto fuori dallo scopo dell'attività.

3.5.2 Exchange Cluster

E' stata rilevata la presenza dell'utente di sistema *fcarray* (password: *fcarray*), probabilmente creato in maniera automatica dall'applicazione omonima. Questo utente risulta avere privilegi amministrativi sulla macchina, e permette di avere accesso in maniera interattiva ai nodi del *cluster* (tramite VNC e Microsoft-RDP) e alle risorse condivise (*share* di rete).

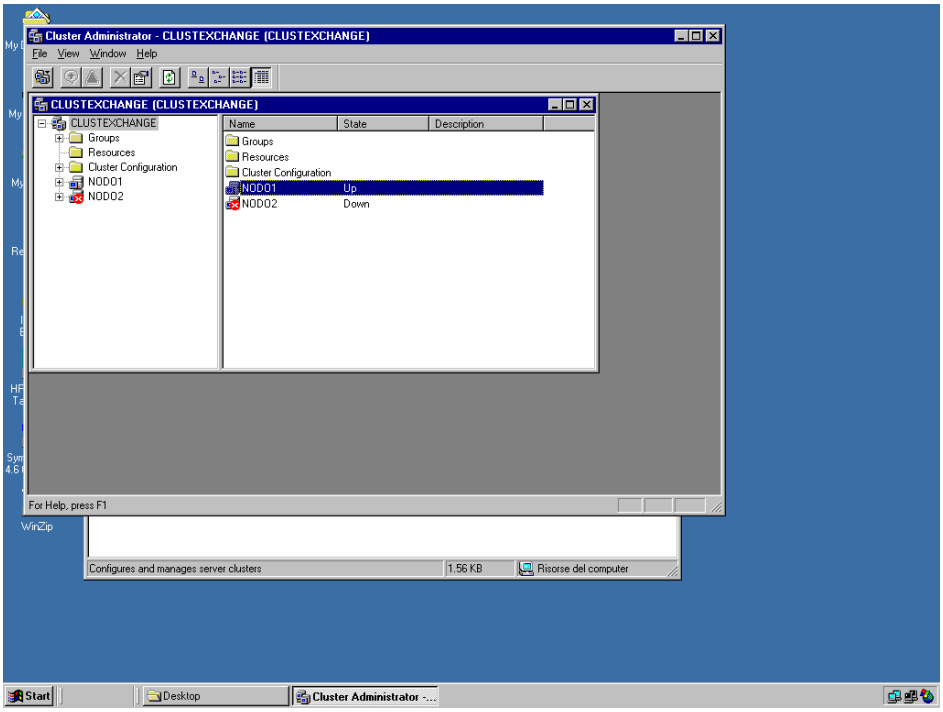


Figura 3 – Interfaccia di amministrazione dei nodi del cluster

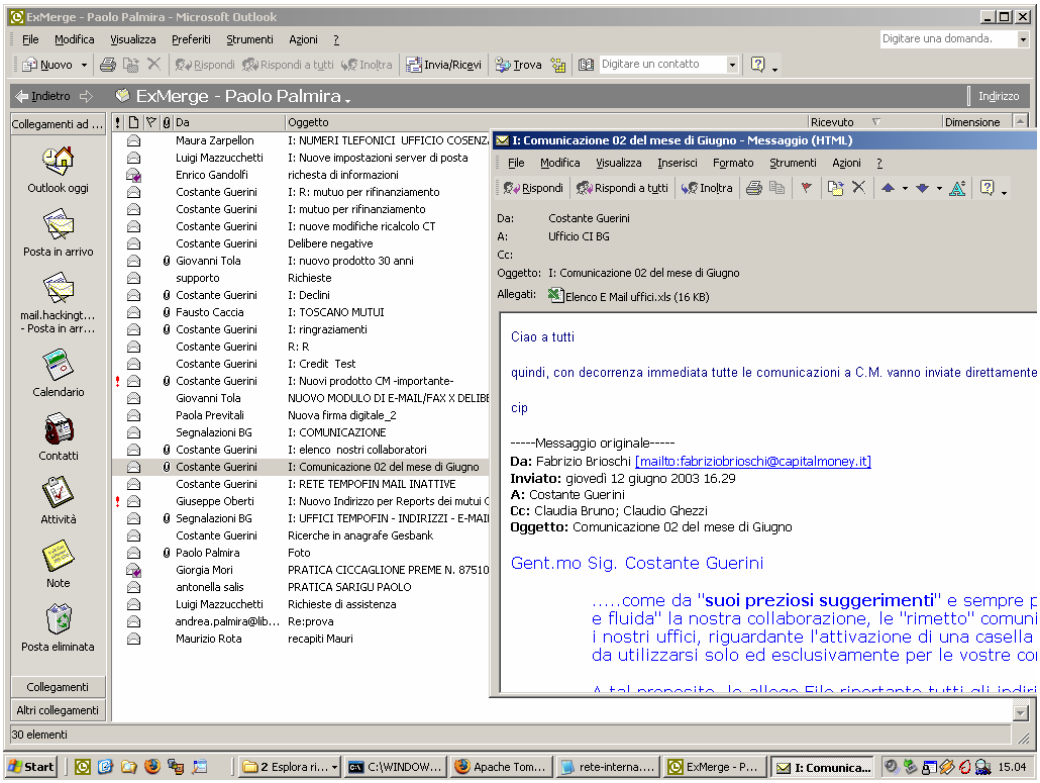


Figura 4 – Una delle mailbox reperite sul server

3.6 Sistema di SSO

L'accesso alle applicazioni intranet servite dal server *appl2* [10.208.17.24] è regolato da un sistema di *Single Sign On* (oracle based) che fa riferimento al server *infra247* [10.208.17.21]. Questo sistema di SSO permette di verificare le credenziali di dominio, che l'utente immette al momento del logon sul proprio client, e che vengono inviate automaticamente dal *browser* (internet explorer) quando si tenta di accedere alle applicazioni web intranet gestite da appl2. In questo modo l'utente, una volta effettuato il logon al proprio client, non dovrà immettere alcuna password ulteriore per accedere ai servizi intranet.

Il sistema di SSO ha un proprio *repository* per le credenziali degli utenti che viene replicato dall'*active directory* contenente le utenze del dominio. Questo sistema permette di uniformare le credenziali (l'utente non deve memorizzare due password) e di applicare al sistema di SSO le stesse policy utilizzate per le password di dominio (es: lunghezza minima). Lo svantaggio di questo tipo di approccio consiste nel fatto che una eventuale compromissione del server che gestisce le credenziali del SSO porterebbe ad una compromissione delle utenze del dominio, e viceversa.

Eventuali vulnerabilità specifiche del protocollo utilizzato per effettuare il SSO non sono state analizzate, in quanto si tratta di una *feature* di un sistema standard commerciale (OracleAS).

Sul sistema che effettua il SSO (10.208.17.24) è stata infine riscontrata la presenza di un portale web che permette l'inserimento di *username* e *password*. Non è stato possibile verificare se tale form di accesso permetta di effettuare il SSO verso le applicazioni intranet, ma il protocollo utilizzato da esso risulta essere HTTP (clear text). Di conseguenza qualsiasi coppia di credenziali inviata in questo modalità può facilmente essere oggetto di intercettazione (vedere paragrafo 3.4).
Tale form di accesso risulta inoltre vulnerabile ad attacchi di tipo *Cross Site Scripting.* Tuttavia, dato il contesto di utilizzo, l'impatto di tale tipologia di attacco è da considerarsi praticamente nullo.

3.7 Applicazioni intranet

Per l'analisi delle applicazioni *intranet* gestite dal server appl2, è stata scelta come campione l'applicazione *GestioneOrdini.* Tale applicazione è stata testata sul server di sviluppo INTEG247-CARTE.
Il test è stato effettuato al fine di evidenziare vulnerabilità che potessero essere sfruttate da un utente legittimo dell'applicazione, mentre il sistema di autenticazione (che viene gestito tramite SSO) non è stato oggetto di analisi in questo senso (vedere paragrafo 3.6). Il Cliente ha quindi fornito delle credenziali valide per accedere all'applicazione e verificare la robustezza delle sue componenti interne (es: possibilità di elevare i privilegi dell'utente, accedere a dati riservati o appartenenti ad altri utenti, etc.).

La logica applicativa che gestisce gli input inseribili dall'utente e' risultata **vulnerabile ad attacchi di tipo SQL Injection** (vedere in proposito il capitolo 4) per la manipolazione arbitraria delle *query* effettuate sul database di *backend*.
Basandosi sul medesimo paradigma implementativo, è possibile supporre che anche le altre applicazioni intranet ospitate sul server siano passibili della stessa vulnerabilità.

4 Riassunto criticità e soluzioni proposte

Di seguito vengono elencate le vulnerabilità *di maggior rilievo*, raggruppate per tipologia, riscontrate durante la fase di analisi, e le relative soluzioni proposte. Per una lista completa delle singole vulnerabilità fare riferimento ai capitoli 2 e 3.

Criticità: SQL Injection.
Sistema: *reports.bankyou.com,* applicazioni intranet
Descrizione: E' possibile, attraverso i *form* di login (*reports*), o interni, eseguire delle interrogazioni SQL sul database utilizzato dall'applicazione (vedere paragrafo 2.2 e 3.7).
Impatto: Un qualsiasi utente su internet (*reports*), o in possesso di credenziali valide (applicazioni intranet), e' in grado di accedere in maniera indiscriminata al contenuto informativo dei database utilizzati dalle applicazioni.
Soluzione: Gli attacchi di tipo *SQL injection*, insieme a molte altre classi d'attacco applicativo (es: *Cross-Site scripting*, *Parameter Tampering*, etc.), possono essere identificati e bloccati con successo da un buon *application-level firewall.* Dispositivi di questo tipo ispezionano i *data-stream* a livello applicativo, identificano determinati *pattern* o comportamenti ritenuti anomali, ed eventualmente bloccano il traffico "incriminato" prima che questo possa giungere al *web server* vulnerabile. Tuttavia, dispositivi di questo tipo sono mediamente costosi e probabilmente sovradimensionati per l'utilizzo in questo contesto (sicuramente nel caso delle applicazioni *intranet*).

In alternativa, si consiglia di rivedere il codice che regola la logica applicativa alla luce delle *best-practice* di programmazione sicura. Nello specifico e' possibile utilizzare, ad esempio, comandi SQL parametrizzati o *stored procedure* per la gestione dei dati forniti dall'utente all'interno delle interrogazioni al database. Si consiglia inoltre di implementare una corretta gestione degli errori generati dal *back-end* (database SQL), per evitare che i messaggi restituiti all'utente possano contenere informazioni utili ad un attacco (es: errori di *data type* delle colonne).

Per maggiori dettagli ed esempi:
`http://msdn.microsoft.com/msdnmag/issues/04/09/SQLInjection/default.aspx`

Criticità: Traffico non cifrato.

Descrizione: Molti protocolli utilizzati nella rete inviano i dati (es: credenziali d'accesso) in chiaro.

Impatto: A causa della mancata presenza di sistemi contro l'intercettazione del traffico, è possibile catturare dati sensibili, e credenziali di accesso ai servizi, mentre essi transitano sulla rete.

Soluzione: Sono possibili due differenti approcci alla risoluzione di questa problematica (non mutuamente esclusivi):

- **Contromisure all'intercettazione del traffico:** Esistono numerosi software per il monitoring e la rilevazione di attacchi volti all'intercettazione del traffico. Tali software possono essere installati come sonde *network* passive, o sonde client attive/passive. Questo tipo di software, tuttavia, protegge dai più comuni attacchi di intercettazione, ma risulta scarsamente efficace nel caso di attacchi evoluti. L'efficacia di questi prodotti inoltre e' direttamente proporzionale alla sua difficoltà di *deployment*.

  Un altro tipo di approccio per mitigare l'impatto di un attacco di questo tipo consiste in una corretta suddivisione (VLAN) della rete nell'ottica di creare zone *trusted* e *untrusted*, e di impedire il passaggio di dati sensibili all'interno di segmenti di rete considerati non "fidati" (quelli ad esempio a cui possono avere accesso i consulenti).

  La soluzione migliore contro questo tipo di attacchi consiste, tuttavia, nell'utilizzo di particolari tecnologie (es: *DHCP Snooping*, *Dynamic ARP Inspection*, etc.) implementate da Cisco direttamente all'interno degli apparati di rete (*switch* e *router*). Qualora il cliente disponesse già degli apparati che supportano questo tipo di tecnologia, Hacking Team potrebbe fornire la consulenza necessaria per una corretta configurazione delle loro *feature* più avanzate, nell'ottica di proteggere questo scenario di rete.
- **Forzare la cifratura dei dati o del canale:** Utilizzare, quando possibile, le funzionalità di cifratura del traffico di determinati servizi (es: HTTPS invece di HTTP, POP3S invece di POP3, etc.). In alternativa e' sempre possibile creare dei canali cifrati punto-punto (es: tunnel SSL) per la protezione di flussi di traffico particolarmente sensibili.

Criticità: Common passwords.

Descrizione: Alcune macchine e servizi presentano utenze di default.

Impatto: E' possibile avere accesso a un discreto numero di macchine e servizi utilizzando password di default. In alcuni casi, queste utenze consentono un accesso con elevati privilegi al contenuto informativo dei sistemi, o addirittura il controllo completo della macchina.

Soluzione: Modificare le *password* relative alle utenze create di *default* in seguito all'installazione di determinate applicazioni (es: SQL Server, Oracle, etc).