UNIVERSITÀ POLITECNICA DELLE MARCHE

DII
Dipartimento di Ingegneria dell'Informazione

unimc

Test security
Vulnerability assessment e penetration testing
verifica infrastruttura interna

Marco Maccari
Unicam Area infrastrutture e Servizi Informatici (AINF)
marco.maccari@unicam.it

Giovedì 2 Ottobre 2025

Corso di Perfezionamento in Cybersecurity, Cyber Risk and Data Protection

Introduzione

Lo scopo di questa attività è stato quello di testare il livello di sicurezza informatica dell'Università di Camerino alfine di simulare un eventuale attacco di un utente che volesse superare le difese perimetrali.
Quindi, è stata effettuata una prova di vulnerability assessment e di penetration test per verificare le eventuali vulnerabilità dei sistemi o l'esistenza di falle che potrebbero essere sfruttate per un accesso non autorizzato, o altra attività dannosa, tale da compromettere la sicurezza dei sistemi e dei dati. Il tutto è stato fatto in collaborazione con una ditta specializzata in test di sicurezza.

Vulnerability Assessment e Penetration Testing

Per la verifica del test di sicurezza sono state svolte entrambe le attività:

Vulnerability Assessment tradizionale/classico

Identificazione e misurazione delle vulnerabilità (discovery & scanning)

Analisi dello stato della sicurezza (result analysis)

Individuazione delle debolezze e definizione delle misure di mitigazione (remediation)

Obiettivo: eliminare o ridurre i rischi a un livello accettabile

Penetration Testing

Simula le azioni di un attaccante esterno/interno
Obiettivo: violare la sicurezza, accedere a dati sensibili o interrompere i servizi
Ethical hacker usa strumenti e tecniche avanzate per tentare il controllo dei sistemi critici

Finalità principali

Individuare vulnerabilità di sicurezza
Valutare efficacia delle policy di sicurezza
Verificare conformità normativa
Misurare consapevolezza dei dipendenti
Testare capacità di rilevare e rispondere a incidenti

PT vs VA

	Penetration Testing	Vulnerability Assessment
Obiettivo	Simulare un attacco informatico per valutare se le vulnerabilità possono essere sfruttate e fino a che punto un attaccante potrebbe comprometterne la sicurezza.	Identificare, classificare e assegnare una priorità alle vulnerabilità presenti in un sistema, rete o applicazione.
Approccio	Include attività di exploit per verificare l'impatto reale di una vulnerabilità.	Analizza l'infrastruttura alla ricerca di vulnerabilità note, senza tentare di sfruttarle.
Automazione	Richiede un intervento umano significativo per identificare exploit non rilevabili da strumenti automatici.	È principalmente automatizzato tramite scanner di vulnerabilità.
Profondità	Valuta le vulnerabilità in un contesto realistico e fornisce una prova concreta del loro impatto.	Identifica le vulnerabilità, ma non verifica se possono essere sfruttate in un attacco reale.
Frequenza	Generalmente è condotto meno frequentemente, spesso su base annuale o in seguito a cambiamenti significativi nel sistema.	Deve essere eseguito periodicamente per mantenere aggiornato lo stato di sicurezza.
Rischio	Può causare interruzioni o crash di sistemi se non eseguito con attenzione.	A basso rischio, poiché non tenta exploit reali.
Costo	Più costoso, dato l'impiego di esperti di sicurezza e test manuali.	Più economico, poiché sfrutta strumenti automatizzati.

Fonte: Gianluca Bonifazi, *Vulnerability Assessment e Penetration Testing, in: slide 30 Maggio 2025, Corso di Perfezionamento in Cybersecurity, UNVPM.*

Scelta del PT

Per la nostra attività è stato scelto come tipo di Penetration Test
Gray Box Testing
Accesso parziale alle informazioni (es. credenziali utente standard, documentazione di rete)
Simula:
attacco di un insider con privilegi limitati
hacker esterno con punto d’appoggio nel sistema

Vantaggi

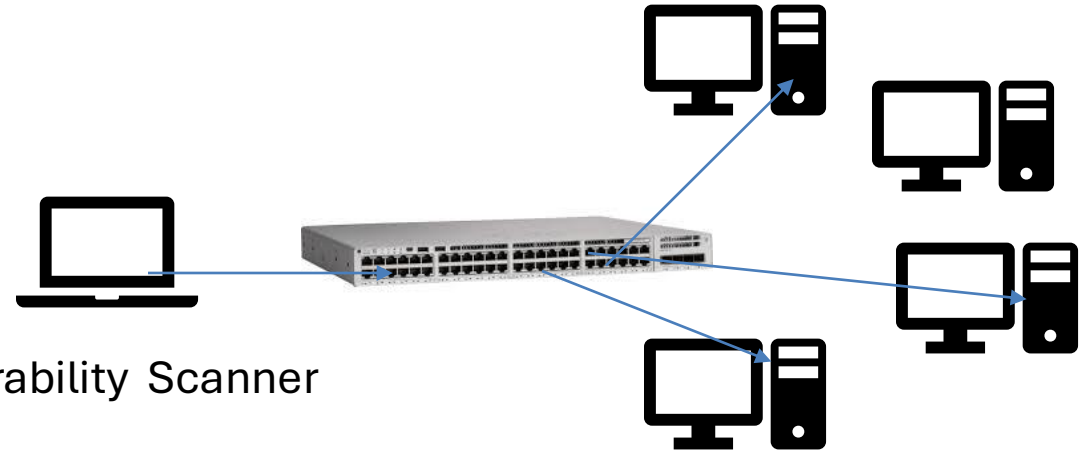
Richiede meno tempo del Black Box
Copertura più approfondita
Svantaggi
Difficile definire i privilegi limitati da assegnare al tester

Vulnerability Assessment
Scansione completa della rete interna

Come strumento è stato utilizzato «Tenable Nessus Professional security scanner»
che ha riscontrato vulnerabilità, confermate con verifiche successive ad hoc su ogni segnalazione

Tenable Nessus Professional security scanner

Lo strumento Nessus è stato installato su una macchina linux virtuale all'interno della infrastruttura Vmware dell'università per simulare la scansione di un eventuale attacco interno.



Nessus Vulnerability Scanner

La scansione ha rilevato versioni di software che si sono rivelate obsolete o non supportate e che devono essere aggiornate.

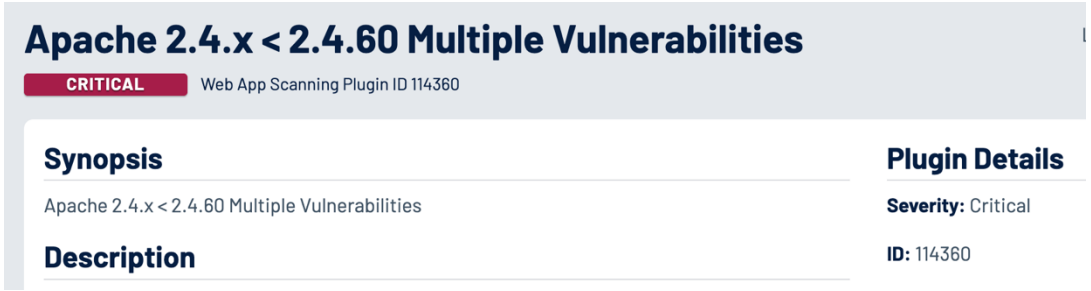
Vulnerabilità rilevate

Vulnerabilità che sono state rilevate

La severità è assegnata in base al valore CVSS v3.1

Bassa (0,1–3,9), Media (4,0–6,9), Alta (7,0–8,9) e Critica (9,0–10,0)

High	Apache Tomcat 11.0.0.M1 < 11.0.6 multiple vulnerabilities
High	Samba Badlock Vulnerability
Medium	MongoDB 6.0.x < 6.0.17 / 7.0.x < 7.0.13 / 7.3.x < 7.3.4 incorrect enforcement of index constraints (SERVER-92382)
Medium	OpenSSH < 10.0 DisableForwarding
Critical	Microsoft RDP RCE (CVE-2019-0708) (BlueKeep) (uncredentialed check)
Critical	Microsoft SQL Server Unsupported Version Detection (remote check)
Critical	OpenSSL 1.1.1 < 1.1.1za Vulnerability
Critical	PHP Unsupported Version Detection
Critical	Redis Server Unprotected by Password Authentication



https://www.tenable.com/plugins/was/114360

Attacco Kerberoasting

Dall’analisi è stata riscontrata la possibilità di un attacco Kerberoasting che avrebbe preso di mira gli account Active Directory

Kerberoasting

- Il Kerberoasting è un attacco estremamente comune negli ambienti di directory attive che prende di mira gli account Active Directory con il set di valori Service Principal Name (SPN).
- Accedendo a un dominio Active Directory come qualsiasi utente autenticato, è possibile richiedere ticket di servizio (TGS) per gli account di servizio specificando il loro valore SPN.
- Se tali SPN sono registrati in Active Directory, il controller di dominio fornirà un'informazione crittografata con la chiave segreta dell'account che esegue il servizio.
- Con queste informazioni, l'attaccante può provare a recuperare la password in chiaro dell'account tramite un attacco di forza bruta.

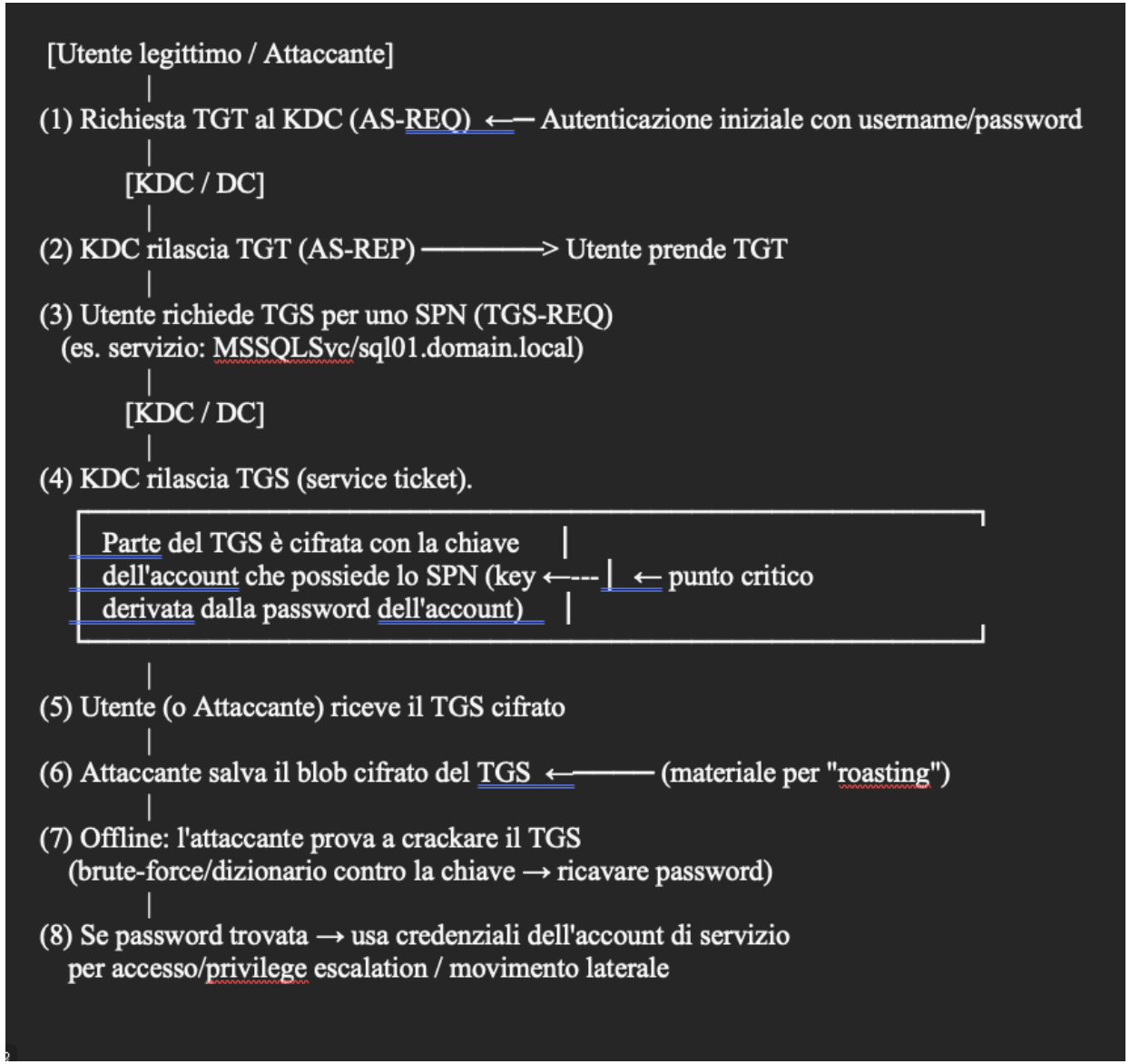
Fasi di un attacco Kerberoasting

- Ricognizione: l'attaccante identifica account di servizio con SPN (Service Principal Name) registrati in AD questi account fanno girare servizi (SQL, HTTP, etc.).
- Richiesta di ticket di servizio (TGS): utilizzando un account valido (anche un account utente compromesso), l'attaccante richiede ticket Kerberos per gli SPN target. Non serve privilegio da amministratore per richiedere questi ticket.
- Estrazione del ticket cifrato: il ticket di servizio ricevuto contiene una parte cifrata che dipende dalla chiave dell'account di servizio (derivata dalla sua password). L'attaccante esporta/usa il ticket.
- Cracking offline: il ticket cifrato viene sottoposto ad attacco offline (brute-force o dizionari) per ottenere la password dell'account di servizio. Poiché l'attacco è offline, non rischia di essere bloccato semplicemente dal sistema target mentre prova password.

Perché questo attacco è efficace?

- Il cracking è offline: non lascia tracce dirette sulle macchine prese di mira durante il tentativo di cracking.
- Molti account di servizio hanno password deboli, non cambiate o usano RC4/NTLM-derived keys, facilitando il cracking.
- Con la password recuperata, l'attaccante può muoversi lateralmente o ottenere privilegi (es. se la password è usata anche su altri host).

Schema attacco Kerberoasting



GetUserSPNs.py

Script Python usato in ambito **Active Directory (AD) / Kerberos**,

in contesto di penetration testing

Lo script In particolare:

- Si connette a un **Domain Controller** tramite LDAP.
- Enumera tutti gli account utente che hanno SPN registrati.
- Richiede ticket Kerberos (TGS) per quegli SPN.
- Restituisce l'hash **Kerberos 5 TGS-REP etype 23 (RC4-HMAC)**, che può essere craccato con tool tipo **Hashcat** o **John the Ripper**.

Esecuzione dello script

L’esecuzione dello script ha prodotto i nomi dei Service Principal associati ad alcuni account utente

MSSQLSvc/QU	Mario.Rossi
MSSQLSvc/ra	Utente1
HTTPS/sm2vcen	Mario.Rossi
HTTP/sm2vcent	Mario.Rossi
HTTP/sm2vcent	Mario.Rossi
MSSQLSvc/arch	Mario.Rossi

Richiesta TGS

Utilizzando le credenziali dello user di dominio della macchina virtuale

sono stati richiesti i TGS

Figure 19 – Kerberoasting

L'hash dell'account «Utente1» è risultato crackable utilizzando dizionari e regole pubblici e ben noti, con conseguente acquisizione della password full cleartext dato che questa password non era abbastanza complessa da essere resistente al tempo a tale attacco.

Fortunatamente l'account non aveva privilegi elevati

Come difendersi

- Forzare l’uso di AES per i service tickets e disabilitare RC4 quando possibile — AES rende il cracking offline molto più costoso
- Usare Managed Service Accounts / Group Managed Service Accounts o soluzioni di gestione delle credenziali che ruotano automaticamente le password. nccgroup.com
- Password forti e lunghe per gli account con SPN (no password statiche, no password brevi).
- Limitare e rivedere gli SPN: rimuovere SPN non necessari o account che non dovrebbero fungere da service account.
- Monitoraggio e rilevazione: anomale richieste di TGS (volume insolito di richieste per SPN o richieste da host non usuali) possono indicare attività di Kerberoasting; allerta su pattern insoliti e correlazione con altri segnali.
- Principio del least privilege / PAM: ridurre il numero di account con privilegi, usare accessi giustificati e temporanei (just-in-time).

Attacco Redis con connessione senza autenticazione

La connessione ad un data base come Redis , senza autenticazione, costituisce una grave debolezza del sistema

Attacco Redis

Sono stati trovati due Remote DIctionary Server (Redis)

Archivi di strutture di dati open source in memoria con accessi non protetti dall'autenticazione con password porta 6379/tcp.

Redis è un archivio dati in memoria comunemente utilizzato per la memorizzazione nella cache, l'analisi in tempo reale e l'intermediazione di messaggi.

- Per impostazione predefinita, Redis non applica l'autenticazione a meno che non sia esplicitamente configurata.
- Un'istanza Redis non autenticata espone il sistema a diversi rischi critici per la sicurezza:
- Esposizione dei dati
- Gli aggressori possono leggere e modificare i dati memorizzati, potenzialmente accedendo a informazioni sensibili.
- Esecuzione remota del codice (RCE) – Se Redis viene eseguito con privilegi elevati, un utente malintenzionato può sfruttarlo per scrivere file dannosi (ad esempio, chiavi SSH, lavori cron) e ottenere un accesso persistente. Interruzione del servizio.

Gli attori dannosi possono emettere comandi FLUSHALL o SHUTDOWN, causando una completa perdita di dati o tempi di inattività del servizio.

Comando redis-cli -h IP del server -p 6379

Per confermare l'accesso non autenticato, è stato eseguito il seguente comando

redis-cli -h IP del server -p 6379

Questo comando apre un client Redis verso l'host sulla porta 6379 (porta predefinita Redis)

Se la connessione va a buon fine e non viene chiesta autenticazione prima di poter eseguire comandi, significa che il server accetta connessioni senza password, accesso non autenticato

Si consiglia di abilitare la connessione autenticata

Conclusioni

L’attività di Penetration Test ha permesso di simulare scenari di attacco realistici al fine di valutare la robustezza delle difese implementate.

L’analisi ha evidenziato:

- la presenza di alcune vulnerabilità e configurazioni migliorabili che, se sfruttate da un attaccante, potrebbero compromettere la riservatezza, l’integrità o la disponibilità delle informazioni;
- aree in cui i controlli di sicurezza hanno risposto correttamente, dimostrando un livello di protezione adeguato contro minacce comuni;
- la necessità di rafforzare specifici aspetti della sicurezza (gestione delle patch, configurazioni, controlli di accesso, monitoraggio).
- In generale, il test ha confermato l’importanza di mantenere un approccio proattivo e continuo alla sicurezza, integrando attività periodiche di verifica tecnica con processi organizzativi di gestione del rischio.

Conclusioni

Abbiamo imparato che è necessario:

- Correggere le vulnerabilità identificate secondo le priorità indicate nel report tecnico.
- Aggiornare procedure e configurazioni di sicurezza, in particolare negli ambiti emersi come critici.
- Ripetere periodicamente attività di Penetration Test e Vulnerability Assessment per monitorare l’efficacia delle misure implementate e adeguarsi all’evoluzione delle minacce.
- Sensibilizzare il personale e rafforzare le misure organizzative a supporto della protezione tecnica.

Conclusioni

L'unica vera sicurezza è sapere di non essere sicuri.

Il paradosso del Penetration test: rompiamo per proteggere, attacchiamo per difendere.

Socrate sarebbe fiero! 😉

GRAZIE PER L'ATTENZIONE!

marco.maccari@unicam.it