



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 al fine 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
<u>Obiettivo</u>	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.
<u>Approccio</u>	Include attività di exploit per verificare l'impatto reale di una vulnerabilità.	Analizza l'infrastruttura alla ricerca di vulnerabilità note, senza tentare di sfruttarle.
<u>Automazione</u>	Richiede un intervento umano significativo per identificare exploit non rilevabili da strumenti automatici.	È principalmente automatizzato tramite scanner di vulnerabilità.
<u>Profondità</u>	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.
<u>Frequenza</u>	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.
<u>Rischio</u>	Può causare interruzioni o crash di sistemi se non eseguito con attenzione.	A basso rischio, poiché non tenta exploit reali.
<u>Costo</u>	Più costoso, dato l'impiego di esperti di sicurezza e test manuali.	Più economico, poiché sfrutta strumenti automatizzati.



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

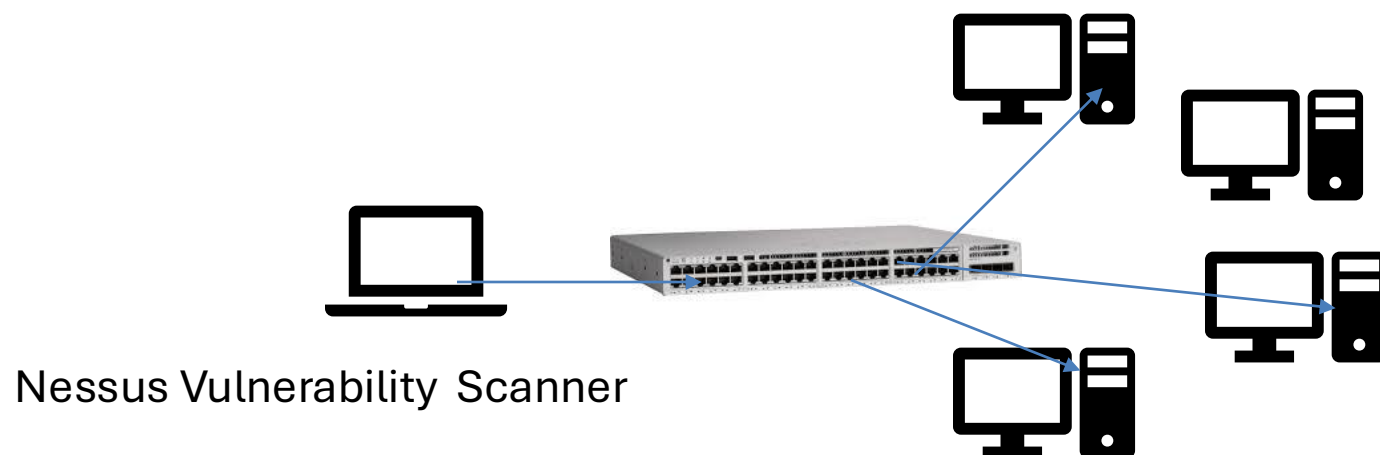
Come strumento è stato utilizzato «Tenable Nessus Professional security scanner» che ha riscontrato vulnerabilità, confermate con verifiche successive ad hoc su ogni segnalazione



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.



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

Apache 2.4.x < 2.4.60 Multiple Vulnerabilities

CRITICAL

Web App Scanning Plugin ID 114360

Synopsis

Apache 2.4.x < 2.4.60 Multiple Vulnerabilities

Description

Plugin Details

Severity: Critical

ID: 114360

<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



[Utente legittimo / Attaccante]

(1) Richiesta TGT al KDC (AS-REQ) ← Autenticazione iniziale con username/password

[KDC / DC]

(2) KDC rilascia TGT (AS-REP) → Utente prende TGT

(3) Utente richiede TGS per uno SPN (TGS-REQ)
(es. servizio: MSSQLSvc/sql01.domain.local)

[KDC / DC]

(4) KDC rilascia TGS (service ticket).

Parte del TGS è cifrata con la chiave
dell'account che possiede lo SPN (key ← ← punto critico
derivata dalla password dell'account)

(5) Utente (o Attaccante) riceve il TGS cifrato

(6) Attaccante salva il blob cifrato del TGS ← (materiale per "roasting")

(7) Offline: l'attaccante prova a crackare il TGS
(brute-force/dizionario contro la chiave → ricavare password)

(8) Se password trovata → usa credenziali dell'account di servizio
per accesso/privilege escalation / movimento laterale

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

<u>MSSQLSvc/QU.</u>	Mario.Rossi
<u>MSSQLSvc/ra </u>	Utente1
<hr/>	
<u>HTTPS/sm2vcen</u>	Mario.Rossi
<u>HTTP/sm2vcent</u>	Mario.Rossi
<u>HTTP/sm2vcent</u>	Mario.Rossi
<u>MSSQLSvc/arch</u>	Mario.Rossi

Richiesta TGS



Utilizzando le credenziali dello user di dominio della macchina virtuale sono stati richiesti i TGS

```
49cac518ac0f358bd80df1bc36d454faff52bfd72712e4fad498868c802c75c1b0ffdb2328d420ea18dbced44601182c3a3b9cff0e98a68df09928298f713010c3f4ceb3c9ea4
145e27e2deea913587165731aab9ee120126f3796455707c75ae1ebf90e4c55c1afababa4e193e048aefc2f4d76b172225bdb71614d9d59fc228fed05af414c9f3e1839a589dbb7c
49902265b7af714822fc5c8aebf979b4f4fc7cba81d07b09d951b3c0702cf240c169ed582c07b146a00fc8d9d21a9215455037c413f6c212dfda923b9a76f42527c4662e892bd22
$krb5tgs$23$<redacted> mario rossi <redacted> j*$c73601e95fdb413fa9af04c2bbaa592a$0a819b64c46cee53bce62210b2d2dcd
cfdbc0fc7f62b2ee50cd627ffd768ddf8b2f4d45f92b50887177daf3dbd19e96dbf1728f98ac07cd9495038fa02310b4c6bc12f46e2e6b19fb2e214d697b8845310954c971b3d9d
779a1c2d4905aed78236c79b459737aad2304bbe9d94e333e24bc98586e36501378bcf798b5bb4bdfc83c7eefdb247f9b3dff5ee530fd6dd5026091bf220e78b5114269f1f89cb3
231b574c7daad5213efb690c97a2385ab85ea3c61c1998682c2f8fe93beb2bcf5dd65a72205575a98e924a83a52bfff3eded1b36b0012b654f8c0e4ee122bcb1a8b1c5d1fdb95d6f
c8b3146eddbdf7fc319d53f0b5310525523b3c14eda252a40701545f70681fd6ea161babb433397e5e77938693f7aefbd23c1fe548bce4c8a2fe4d9fb6fa248032818a2d806730f
ae2b3dc1263093d760ed4e08b03974164c6c7a0852d1a9a299214a0d8565e6d01bee03167a78bfa3a886d8adbd248c68af9f3650b7457d5a86d52331e45052a6bcabefa7641e08d
d8892a0b976236f6500efc19e5befb2ff142c8cef56e40a093591f742350c668b08405a690314bdfebc0bee186a485f60d3e2481f405b53aa4e3fffb1c6356eba3fa887eafe67f
a9c3a8324ed9bd7729fe810af8e2cfdaed6ef0466615692d5a5e08248d5952e04a2b4a0bdaef0e63e82a52918f688f6769398d8f5f6668c40f3925d9617d65e69ec550fa7453b1d
d043978ae1c2b3a1c3db537e7423eb341b6aa65adce22fa2c4b49a65e8332a53d88feaa75acddee7668cae0f1009335292959c654b4f6e3d61931998a358409e530d9fac92f1faa
$krb5tgs$23$<redacted> mario rossi <redacted> Testo j*$95f6fb6b606be10ea07d69c03f2d6c5d$bbc47722600a2a39c123978a1185f2a2
43b2422d09d58f899aa4c123e74879b14d4cde917adfb075fd84a01de09a0341df98e639319c81448b6bc39889f7c7be1b9e172439202706c6bc23d411827344d7d6c9e8e70bfe
0966a849471b8bc6d9b907f7a0cafb927a0b58e8a38e8cb918ebd3e5a91c37633253a94a5d27c14390675fd36be94dfbc25eff60c17e81d8737f86f6e31164c08bbfcf8a4c0e3a2
8d21015f56128f97a8a52a163e353c5bcac5d3bfd223fb54d0702f73c9f5d537d65dde3fe7411eb3fd90ff97481c7b19148ac9d59aca8e565cc11d14d67c408cc3bdcf73497f9fd
ef5d9ffe388bee0360338c057644fe228eda1a9b3d36ae385ad9f74179a4ea93eb119346dace7c5df45ce8d41e6875c42c324fe2b13199bc70e840f630fdbcd0074176d3c0742ca
bd3bb97869dbff1f7c8a3ddfc52441564e57277e4da08c1d1d3154d65094d24f5781a588d06fbcc0cef8777d51a101c64a410aba180f100cd426bb2d2dd90905f93b109d06f7442
f57aafd7afc4947a977f02ebe997670a7c0eaf74a6051e5eaf2fa22409d994018f198c286c2301ff4d0f5bb9d0a272813248e675293757ab3de90e584c142dd8007d6b606f1fc8e
eaad524427a29f72e3d1359e89ac8d217d3455024cf24b56eb0fb1cacc3a931c858a770a3c42d2b76e86b40aff24af465094ed8ef203e9104c5708bb13256a84b9e3670e82cedd
ce5ae5265fb8f9503a350f8c5f50c096f95d907427618ffa685d84099e8fe000a3979a79c077236387db6a351064225f954b054adc6c53b2e2fa2e27d98de1837d6c1bb316681b4
$krb5tgs$23$<redacted> Utente 1 <redacted> jcb0c3589af6727fb5bf9c334dd$1e12fdae623b0b7d42ec0e936c6a3d21af2f98
bef02b11b5ccf57b0b451e5eb8e58704adfd90302257c69c32f98367d6a7a6d7017fb8655a72b31d4995c0cc17e8e5ec392b4b7926ee0d27112cde0f72d31a54dcea64d3f5e032
```

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