PROGETTO 2

DECEPTION COMPONENT GENERATOR

Gloria Pelucchini

Leonardo Lembo

Stefano Paparella

Alessio Troffei

GENERATORE DI RISORSE INGANNEVOLI

In ambito di sicurezza informatica l'asimmetria informativa viene utilizzata come leva strategica sia dagli attaccanti che dai difensori, perché è sempre possibile avere un accesso più approfondito a delle informazioni dettagliate rispetto all'altra parte. Gli attaccanti, infatti, la sfruttano per studiare la struttura difensiva del sistema, captando le vulnerabilità e le aree più deboli in modo da progettare attacchi mirati e concentrare tutte le risorse sui punti deboli. I difensori invece, possono utilizzare l'asimmetria difensiva per diffondere informazioni false sui loro sistemi di sicurezza in realtà non attivi. Un esempio sono gli "honeypot" ovvero risorse fittizie che sembrano autentiche ma il cui unico scopo è quello di attrarre gli attaccanti e quindi indirizzarli in zone del sistema meno critiche andando anche a configurare false vulnerabilità per deviare la loro attenzione.

Gli esperti informatici hanno proposto negli ultimi anni di utilizzare "l'inganno difensivo" per ribaltare l'asimmetria informativa ingannando gli attaccanti per proteggere un sistema. Vengono creati servizi, componenti o risorse false in grado di attirare l'attenzione degli stessi, che dispendono forze per studiare un attacco alle risorse fittizie credendo che siano importanti e sensibili, quando invece servono proprio per salvaguardare e difendere ciò che è effettivamente di valore. I motivi dell'utilizzo di questa tecnica possono essere molteplici, introducendo risorse false (ovvero "miele") si possono sia distogliere gli attaccanti da obiettivi reali, ma si possono anche individuare attività sospette prima del verificarsi di danni veri e propri o anche solo raccogliere informazioni sull'approccio e tecniche utilizzate dagli aggressori. Questo può essere un punto di partenza per il futuro di tante realtà aziendali on evitare danni futuri e sensibilizzare gli addetti al campo sicurezza.

Il nostro progetto si impone di creare un sistema di honeypot con il file Server SAMBA, arricchito con l'autenticazione LDAP ed altre funzionalità volte a creare un substrato solido per il componente deceptive.

Il file server SAMBA è una suite di programmi open-source che semplifica la condivisione di file e risorse tra sistemi operativi diversi (presenti sulla stessa rete) e soprattutto va ad implementare il protocollo SMB/CIFS (Server Message Block/Common Internet File System), che è comunemente utilizzato nei sistemi Windows. L'utilizzo di SAMBA per creare questo sistema è dato dal fatto che è si tratta di un tool ampiamente diffuso e conosciuto dalla comunità IT e gli aggressori saranno sicuramente più inclini ad esplorare e tentare di compromettere risorse che utilizzano tecnologie comuni e diffuse. Un'altra ragione è la modularità e configurazione le quali permettono di rendere più convincente una risorsa falsa anche simulandone il comportamento, inoltre, è anche possibile impostare un monitoraggio degli accessi e attività che permette a chi detiene il sistema di raccogliere informazioni dettagliate, e quindi anche autorizzare un'analisi delle tattiche degli aggressori, cercando di migliorare le difese del sistema proteggendo le proprie vulnerabilità.

Per questo progetto abbiamo utilizzato le seguenti guide:

- Per l'installazione e la configurazione di Samba: https://ubuntu.com/tutorials/install-and-configure-samba#1-overview
- Per effettuare il collegamento tra Samba e LDAP: https://docs.huihoo.com/ubuntu/11.10/serverguide/samba-ldap.html

COLLEGAMENTO SAMBA-LDAP

Per poter implementare una architettura software sufficientemente elaborata come quella tra una base dati LDAP (sottolineando che LDAP non è altro che un protocollo di accesso) e un file server SAMBA sono necessari molteplici servizi attivi.

Installazione di LDAP

Per quanto riguarda l'installazione di LDAP, sono necessari i tool:

```
sudo apt install slapd
sudo apt install ldap-utils
```

Slapd è il demone di sistema incaricato ad avviare il server directory LDAP basato su openLDAP.
Per potervisi interfacciare è necessario il package `ldap-utils` il quale implementa tutte le operazioni di base per poter effettuare query, aggiungere, cancellare e modificare le entries della directory.
Dopo di che, è necessario configurare il server LDAP. Ciò è stato fatto attraverso il comando

```
sudo dpkg-reconfigure -plow slapd
```

Non appena si avvia questo comando, verrà mostrata la schermata di configurazione.
Bisognerà innanzitutto inserire:

- Il DNS domain name, che andrà a costituire il Distinguish Name del root LDAP della nostra gerarchia;
- Il nome dell'organizzazione che rappresenta un campo informativo riguardante il DIT (Directory Information Tree).

Per semplicità entrambe le voci sono state impostate al valore "`cyber.samba.org`", creando quindi l'omonimo dominio "`dc=cyber,dc=samba,dc=org`".
Successivamente viene chiesta la password dell'amministratore (in questo caso si è scelto "secret") la cui entry si trova sotto "cn=admin" seguito dal suffisso LDAP scelto al punto precedente; infine, come impostazioni facoltative, si può configurare il servizio in modo tale che il database venga eliminato al momento della rimozione di slapd e si può scegliere di importare il database precedentemente presente nel sistema.
Per poter creare una struttura gerarchica compatibile con il file server samba è stato necessario importare uno schema LDAP all'interno del nostro servizio.
Per fare un esempio, consideriamo una entry di un utente al quale vengono associate una serie di proprietà: username, home directory, user password, ecc.
Queste proprietà vengono dette **attributi**, e tali attributi sono definiti da uno **schema** interno alla struttura dati. Ciascun profilo rappresenta pertanto un singolo elemento e i suoi attributi sono definiti dallo **schema**, che determina anche il tipo di oggetti che possono essere immagazzinati all'interno della nostra Directory LDAP.
Lo schema può essere trovato all'interno del package `samba-doc` e deve essere opportunatamente installato per poter funzionare correttamente.

Dopo aver preparato i seguenti due file:

schema_convert.conf

```
include /etc/ldap/schema/core.schema
include /etc/ldap/schema/collective.schema
include /etc/ldap/schema/corba.schema
include /etc/ldap/schema/cosine.schema
include /etc/ldap/schema/duaconf.schema
include /etc/ldap/schema/dyngroup.schema
include /etc/ldap/schema/inetorgperson.schema
include /etc/ldap/schema/java.schema
include /etc/ldap/schema/misc.schema
include /etc/ldap/schema/nis.schema
include /etc/ldap/schema/openldap.schema
include /etc/ldap/schema/pmi.schema
include /etc/ldap/schema/samba.schema
```

Configurazione che afferma quali schemi LDAP prendere in considerazione

samba_indices.ldif

```
dn: olcDatabase={1}mdb,cn=config
changetype: modify
add: olcDbIndex
olcDbIndex: loginShell eq
olcDbIndex: uniqueMember eq,pres
olcDbIndex: sambaSID eq
olcDbIndex: sambaPrimaryGroupSID eq
olcDbIndex: sambaGroupType eq
olcDbIndex: sambaSIDList eq
olcDbIndex: sambaDomainName eq
olcDbIndex: default sub
```

File che contiene le modifiche da apportare alla configurazione della directory LDAP

I passaggi che devono essere seguiti per importare il nuovo schema:

1. Spostare lo schema nella cartella `/etc/ldap/schema`
2. Tramite il comando:

```
slapcat -f schema_convert.conf -F ldif_output -n0 -H
ldap:///cn={12}samba,cn=schema,cn=config -l cn=samba.ldif
```

   Per generare all'interno della cartella `ldif_output` le configurazioni che rispettano gli schemi inclusi nel file `schema_convert` e per poi recuperare il LDAP Directory interchange format (LDIF) per lo schema di samba generato `cn={12}samba,cn=schema,cn=config`
3. l'output di slapcat è destinato ad essere utilizzato come input per slapadd. L'output di slapcat ge neralmente non può essere utilizzato come input per ldapadd, motivo per il quale bisogna appor tare piccole modifiche al file `cn=samba.ldif`
4. `ldapadd -Q -Y EXTERNAL -H ldapi:/// -f cn\=samba.ldif`
   con il quale aggiungiamo lo schema samba con `dn: cn=samba,cn=schema,cn=config` all'interno della configurazione della directory LDAP
5. `ldapmodify -Q -Y EXTERNAL -H ldapi:/// -f samba_indices.ldif`

Installazione di un servizio di naming per LDAP

Finita questa serie di passaggi ed aver preparato il server LDAP a fornire utenti in formato POSIX per gli utenti S AMBA c'è la necessita di rendere disponibili queste entries tramite un servizio di naming
A tal proposito è stato installato tramite il comando:

```
apt install -y libnss-ldapd
```

Demone che configura un servizio NSS (Name Service Switch) che permette di esporre e rendere interrogabili f onti di informazioni del sistema come utenti, gruppi e password, indipendentemente dalla sorgente da cui ques ti dati provengano (Nel nostro caso LDAP)
Andando a definire l'indirizzo delle risorse come `ldapi:///`
e LDAP server search base come `"dc=cyber,dc=samba,dc=org"`
si termina la configurazione del collegamento tra samba e LDAP mettendo tra le fonti del file `/etc/nsswit ch.conf` il servizio LDAP nel seguente modo.

```
passwd:         files ldap
group:          files ldap
shadow:         files ldap
```

Per testare il corretto funzionamento si può utilizzare il comando `getent passwd` (dopo aver aggiunto ute nti validi al server LDAP).

Installazione di Samba

Per installare il File Server Samba, abbiamo prima installato il relativo package:

```
sudo apt install samba
```

Successivamente, abbiamo installato il tool *smbldap-tools*, necessario alla creazione e alla gestione degli utenti e dei gruppi LDAP e al loro accesso al FS Samba:

```
sudo apt-get install smbldap-tools
```

Conclusa l'installazione abbiamo configurato i vari componenti LDAP, per completare il collegamento con Samba, con il comando

```
sudo smbldap-config
```

che mostra all'utente il form di configurazione. Nello specifico, sono state selezionate le seguenti opzioni:

- **SID** del dominio → è stato inserito lo stesso SID ottenuto col comando `net getlocalsid`.
- **sambaDomain** → abbiamo lasciato il valore predefinito, cioè "WORKGROUP".
- **SlaveLDAP** e **MasterLDAP** → sono rispettivamente le informazioni di client e server di LDAP, entrambe contenenti l'indirizzo IP del client/server e il Distinguish Name dell'admin di ognuno. In questo caso, entrambi hanno IP pari a 127.0.0.1 e DN pari a `"cn=admin,dc=cyber,dc=samba,dc=org"`.
- Impostazione dei suffissi, per completare i vari DN del nostro server Samba con autenticazione LDAP:
    - **suffix** → suffisso generale presente in ognuna delle voci sottostanti, fissato a `dc=cyber,dc=samba,dc=org`;
    - **groups** → denomina i gruppi Samba, impostato come `ou=groups`;
    - **users** → denomina gli utenti Samba, impostato come `ou=users`;
    - **machines** → denomina le macchine Samba, impostato come `ou=computers`.

Una volta configurati tali componenti, abbiamo proceduto al loro inserimento nello schema di Samba, tramite il comando

```
sudo smbldap-populate
```

In questo modo, il server LDAP ha tutte le informazioni necessarie per autenticare gli utenti di Samba.
L'ultimo passo di questo processo di configurazione è quello di configurare Samba, in modo tale che accetti l'autenticazione tramite LDAP. Abbiamo raggiunto questo obiettivo inserendo, nel file `/etc/samba/smb.conf`, le seguenti opzioni:

```
passdb backend = ldapsam:"ldap://127.0.0.1"
ldap suffix = dc=cyber,dc=samba,dc=org
ldap user suffix = ou=users
ldap group suffix = ou=groups
ldap machine suffix = ou=computers
ldap idmap suffix = ou=idmap
ldap admin dn = cn=admin,dc=cyber,dc=samba,dc=org
ldap ssl = no
ldap passwd sync = yes
add machine script = sudo /usr/sbin/smbldap-useradd -t 0 -w "%u"
```

Queste impostazioni servono a comunicare a Samba l'indirizzo del server LDAP che si occuperà dell'autenticazione degli utenti e di tutte le informazioni necessarie per adeguare il DN degli utenti Samba ammissibili con quello precedentemente impostato per LDAP. Inoltre, abbiamo impostato la sincronizzazione delle password fra i due servizi, si impedisce l'utilizzo di StartTLS, che cifra le comunicazioni che avvengono all'interno del funzionamento di LDAP. Abbiamo fatto ciò per raggiungere lo scopo di avere una macchina che funga da honeypot. Infine, abbiamo aggiunto uno script che, per ogni macchina inserita nel server Samba, crea un utente con lo stesso nome della macchina, impostando come password il nome utente.

Impostazione dello share pubblico e privato

Una volta effettuato il collegamento tra Samba e LDAP, abbiamo configurato lo share pubblico e quello privato, per ogni utente. Per quanto riguarda il primo, abbiamo creato la cartella `/home/public` e abbiamo modificato il file di configurazione di Samba affinché la riconoscesse come share pubblico, con i seguenti comandi:

```
[public]
   comment = Pubblic Directory
   path = /home/public
   public = yes
   read only = no
   browseable = yes
   guest ok = yes
   force user = nobody
   force group = nogroup
```

Per quanto riguarda lo share privato, abbiamo sfruttato la già presente configurazione di Samba `[homes]`, che definisce quali siano le cartelle private di ogni utente. Essa ci ha permesso di usare, come cartella privata di ogni utente, la cartella `/home/<username>`, che viene creata di default al momento della creazione dell'utente stesso.

In questo modo, abbiamo completato la configurazione di Samba, ammettendo anche l'autenticazione tramite LDAP.

SCRIPT DI GENERAZIONE DEL FILESYSTEM FITTIZIO

La generazione del file system fittizio è ottenuta mediante uno script Bash e rappresenta il cuore del progetto nonché la base di implementazione della deception technology. Permette infatti la creazione di una gerarchia di file verosimile per ogni utente e/o gruppo aggiunto all'albero LDAP. È comunque importante capire quali sono le sue funzionalità principali, andandole ad analizzare in maniera approfondita:

- All'avvio dello script è richiesto di inserire in input il numero di utenti da aggiungere all'albero LDAP;
- Per ogni utente, è necessario digitare il nome e il gruppo a cui lo si vuole aggiungere. Come prima cosa viene quindi verificato se l'utente in questione esiste già tramite il comando:
  `ldapsearch -Q -LLL -Y EXTERNAL -H ldapi:/// -b "ou=users,dc=cyber,dc=samba,dc=org" "(uid=nome_utente)"`
  Viene poi fatta la stessa cosa per il gruppo;
- Se l'utente non esiste verrà aggiunto con:
  `smbldap-useradd -m -P -a nome_utente -s ""`
- Se si è scelto di aggiungere un nuovo gruppo per l'utente, allora verrà creato tramite:
  `smbldap-groupadd -a nome_gruppo`
- Nel caso il gruppo è già presente nel sistema, l'utente viene aggiunto a questo gruppo con:
  `smbldap-groupmod -m nome_utente nome_gruppo`
- Ora si vanno a creare e riempire con file fittizi rispettivamente:
  - La cartella privata dell'utente;
  - La cartella del gruppo a cui l'utente è stato aggiunto (quest'ultima verrà riempita soltanto se non fosse già esistente, e quindi non già completa di file). Questo passo si conclude con il settaggio dei permessi per la cartella del gruppo, in modo tale che soltanto gli utenti appartenenti possano modificare i file all'interno.

  La generazione di questi file è affidata alle funzioni:
  - **`gen_folder_name()`:** Per generare il nome della cartella;

    - **`gen_file_name()`:** Per generare il nome del file;
    - **`gen_file_content()`:** Per generare il contenuto del file;
    - **`gen_file_hierarchy()`:** Per generare la gerarchia del filesystem fittizio.

    Ognuna di queste funzioni utilizza dei file esterni:

    - **`folder_names.txt`:** Contiene i nomi possibili delle cartelle;
    - **`file_names.txt`:** Contiene i nomi possibili dei file;
    - **`titles.txt`:** Contiene I titoli dei vari paragrafi contenuti nei file generati;
    - **`paragraphs.txt`:** Contiene il testo dei paragrafi contenuti nei file generati;
- Come ultimo passo, lo script modifica la configurazione del demone di ClamAV (`clamd`) in modo da potere analizzare in tempo reale le cartelle appena create. Così facendo, ogni accesso a file infetti verrà negato e verranno spostati nella cartella di quarantena, evitandone così la diffusione.

PANDOC

Pandoc è uno strumento che permette di convertire documenti da un formato di markup ad un altro in maniera semplice e veloce. Un formato di markup è un linguaggio che usa dei simboli o label per definire la struttura e la formattazione di un testo. Pandoc supporta molti formati di markup, tra cui Markdown, HTML, LaTeX, Word, PDF, ma è compatibile anche con formati proprietari come i documenti di Microsoft Word.

Utilizzo

Pandoc è risultato particolarmente utile in quanto è utilizzato per convertire i file temporanei (generati in formato Markdown) direttamente in documenti PDF, Word (.docx) o in "OpenDocument Text" (.odt). Importante notare che il sistema di generazione dei file fittizi prende informazioni principalmente da:

- **Un file contenente i titoli dei paragrafi:** Qui sono presenti un'ampia varietà di titoli possibili tra cui scegliere, tutti generati utilizzando LLM quali ChatGPT;
- **Un file contenente i paragrafi stessi:** Contenente che verrà scritto in ogni paragrafo. Anche questo è stato scritto aiutandosi con LLM.

Una volta convertiti, i documenti vengono usati per popolare il file server. Più precisamente, si vanno a popolare le cartelle relative agli utenti e ai gruppi in modo da creare una gerarchia tipica presente in un classico PC. I documenti ottenuti sembrano quindi veritieri, ma in realtà non contengono informazioni sensibili e danno così vita ad un ambiente ingannevole per gli attaccanti.

Di seguito sono riportati gli esempi di conversione utilizzati nello script di generazione del filesystem fittizio:

- Per convertire in formato PDF si utilizza il seguente comando:
  `pandoc input.txt --pdf-engine=xelatex -o output.pdf`
- Per convertire invece in formato documento Word:
  `pandoc -s input.txt -o output.docx`
- Infine, per convertire in formato OpenDocument:
  `pandoc input.txt -o output.odt`

CLAMAV

Si è scelto di integrare nel sistema il software "ClamAV" come antivirus in quanto ormai molto affermato nel panorama dell'open-source. Il file server, considerato un potenziale bersaglio degli aggressori, richiede solide misure di sicurezza per proteggere la sua integrità e salvaguardare la credibilità dell'inganno stesso. Il

rilevamento di malware può quindi trarre in inganno gli aggressori, convincendoli di essersi infiltrati con successo in un sistema autentico. Questa confusione può ostacolare l'avanzamento dell'attacco e potenzialmente distoglierlo dalle risorse critiche.

Utilizzo e configurazione

ClamAV è utilizzato per scansionare periodicamente i file delle cartelle condivise. Questo processo sarà programmato per garantire i file vengano sottoposti a verifica regolare, riducendo al minimo il rischio di minacce persistenti. Per ottenere questo comportamento si sono utilizzati tre componenti principali:

1. File di configurazione di clamd (`clamd.conf`)
2. Systemd service file (`clamav-onacc.service`)
3. Script Bash (`detected.sh`)

File di configurazione `clamd.conf`

Il file `clamd.conf` contiene le impostazioni principali per il demone di ClamAV, (ovvero clamd) che è responsabile della scansione in tempo reale. Alcuni settaggi rilevanti sono:

- **`OnAccessPrevention yes`:** Abilita la prevenzione in tempo reale, in modo da impedire l'esecuzione di file infetti;
- **`VirusEvent /etc/clamav/detected.sh`:** Specifica lo script da eseguire quando viene rilevata una minaccia. Questa riga è di fondamentale importanza, in quanto questo specifico script permette di loggare e notificare il tipo di malware trovato;
- **`OnAccessIncludePath /home/user2/Desktop`:** Include il percorso `/home/user2/Desktop` nella scansione in tempo reale.

Systemd service file `clamav-onacc.service`

Questo è un "**systemd service file**" ed è stato creato appositamente per il componente "ClamAV On-Access Scanner", noto come **`clamonacc`**. Questo si occupa di eseguire la scansione in tempo reale. Di seguito è riportata una porzione del file:

```
[Service]
Type=simple
User=root
ExecStart=/usr/sbin/clamonacc -F --config-file=/etc/clamav/clamd.conf --
log=/var/log/clamav/clamonacc.log --move=/home/quarantine
```

Da notare il comando `ExecStart`, che permette di inizializzare il servizio. In questo caso si tratta dell'eseguibile `clamonacc` che viene avviato con determinati parametri, tra cui il file di configurazione (`--config-file`), il percorso del file di log (`--log`), e la cartella di quarantena in cui verranno spostati i file infetti rilevati (`--move`).

Script `scanner.sh`

Questo script, come detto nella configurazione di `clamd`, viene eseguito quando viene rilevato un malware. Le azioni principali che deve svolgere includono:

- La registrazione dell'evento nel file di log.
- L'esecuzione di `clamdscan` per ottenere informazioni sugli elementi infetti.

- Se sono state rilevate delle minacce, si procede nel seguente modo:
    - Si invia messaggio di emergenza al logger di systemd utilizzando `systemd-cat`.
    - Si manda una notifica a tutti gli utenti grafici attualmente connessi utilizzando `notify-send`. Da notare che l'invio di notifiche ai singoli utenti grafici utilizza `notify-send` e richiede che l'utente abbia un server DBus attivo. L'accesso al bus DBus è gestito per ciascun utente grafico individuale.

```
...

sudo clamdscan --log "$LOG" --infected --multiscan --fdpass "$TARGET" > "$SUMMARY_FILE"
...
  if [[ -n $(command -v systemd-cat) ]] ; then
    echo "Virus signature found - $INFECTED_SUMMARY" | /usr/bin/systemd-cat -t clamav -
p emerg
  fi

  XUSERS=($(who|awk '{print $1$NF}'|sort -u))
  for XUSER in $XUSERS; do
  ...
      /usr/bin/notify-send -i security-low "Virus signature(s) found"
"$INFECTED_SUMMARY"
  done
```

Versione container Docker

Nel caso della macchina virtuale, la configurazione appena descritta è completamente funzionante. Purtroppo, date le limitazioni dei container, non è stato possibile implementare questa architettura in maniera completa. Infatti, nei container Docker non è consentito (ed è altamente sconsigliato) l'utilizzo di Systemd come sistema di init. Questo ostacolo è stato comunque superato andando ad utilizzare `crond`. Questa utility permette di pianificare l'esecuzione automatica di comandi o script a determinate date/orari. Per fare ciò è necessario configurare il file `/etc/crontab` specificando il comando da eseguire dettagliandone la data. Questi comandi automatici prendono il nome di "cron job", e sono le attività che crond esegue periodicamente secondo la pianificazione scritta.

Di seguito è riportato lo script (`clamscan.sh`) legato al cron job che verrà poi eseguito periodicamente:

```
#!/bin/bash
LOG_FILE="/var/log/clamscan.log"
QUARANTINE_DIR="/var/quarantine"
mkdir -p "$QUARANTINE_DIR"
clamscan -r /home/ --log="$LOG_FILE" --move="$QUARANTINE_DIR"
echo "Scan completed on $(date)" >> "$LOG_FILE"
```

Tale script verrà eseguito ogni giorno a 0:00. Per far questo è necessario utilizzare `crontab -e` inserendo:

```
0 0 * * * /home/public/clamav_docker/clamscan.sh
```