Verifica robustezza dei servizi di rete:gli scanner

Nessus

Saint

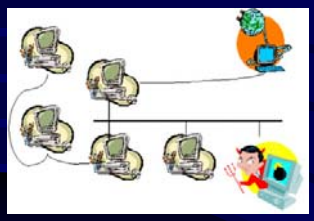
Nmap

Portscan

Istruzioni

Fare clic su uno dei pulsanti nella parte superiore della diapositiva per visualizzare le informazioni relative ai vari argomenti

Introduzione



Tipi di penetrazione:

- •Virus
- •Worms
- •Trojan horses
- •Bombs
- •Backdoor (anche dette Trap doors)
- •Probing del sistema
- •Tecniche di scanning

I Port Scanner esaminati

- Portscan
- Nmap
- Nessus
- Saint

Il nostro obiettivo

Cosa vogliamo ottenere:

- Struttura
- Configurazione

Come l'otteniamo:

- Improntamento
- Scanning

L'improntamento

Cosa riesce ad ottenere:

- Amministratori e utenti della rete
- Posizione del server
- Presenza o meno di Intranet
- Sistemi di rilevamento delle intrusioni
- Server DNS e sua configurazione
- Indirizzi IP assegnati
- Eventuale accesso telefonico

Lo scanning

Cosa riesce ad ottenere:

- Quali sono le macchine attive e raggiungibili
- Quali sono i servizi disponibili

Che tecniche usa:

- Il ping sweep
- Il portscanning
- Il rilevamento del sistema operativo tramite stack TCP/IP fingerprinting

Port scanning

Definiamo come portscanning il processo di connessione a porte TCP e UDP sul sistema nel quale si vuole tentare una penetrazione al fine di determinare quali servizi siano in esecuzione o in stato di LISTENING

Obiettivi dell'intruso

- Abusare dei servizi malconfigurati o soggetti a bug
- Identificare i servizi TCP o UDP in esecuzione
- Identificare il Sistema Operativo
- Identificare applicazioni specifiche
- Identificare versioni di un dato servizio

Protocollo TCP/IP

- Consente un tipo di collegamento connection-oriented
- Protocollo aperto

TCP flag

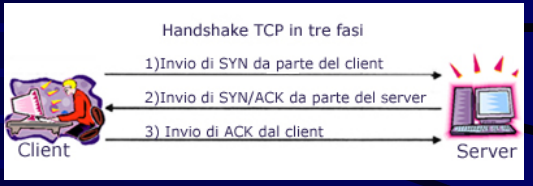
- TCP flag: contiene informazioni vitali per il filtraggio

- SYN
- ACK
- RST
- FIN

Tipi di scansione

- Scan TCP connect()
- Scan TCP SYN
- Scan TCP FIN
- Scan TCP Xmas Tree
- TCP Null Scan
- UDP scan

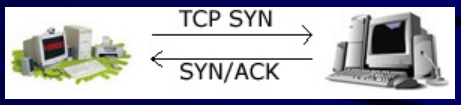
Scan TCP connect()



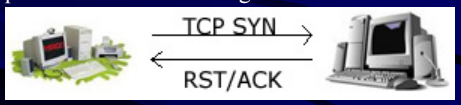
- Facilmente rilevabile dal sistema che lo subisce
- Raramente usato

Scan TCP SYN

- La porta P è in Listening:



- La porta P non è in Listening:



- Non viene realizzata una connessione TCP completa ("half-open scanning")

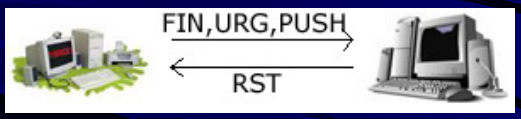
Scan TCP FIN

- La porta P è  :

- Funziona solo su implementazioni UNIX di TCP/IP

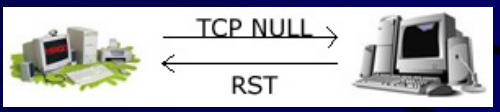
Scan TCP Xmas Tree

- Tutte le porte sono  :



TCP Null Scan

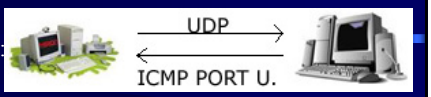
- Tutte le porte sono  :



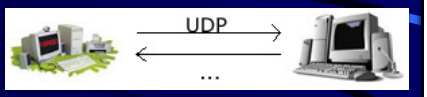
- Disattiva tutti i flag

UDP Scan

- La porta è  :



- La porta è  :



 Protocollo estremamente lento

Confronto tra i software

- Sistemi operativi supportati
- Interfaccia
- Architettura
- Distribuzione
- Ambito di utilizzo
- Tipi di scansioni
- Servizi offerti
- Banco di prova
- Spazio su disco

Tipi di sistemi operativi supportati

Saint	linux, solaris, Sun Os
Portscan	Os2, linux, Win 9x/Nt/2000
Nessus	Lato Server(Unix like), lato Client tutti
Nmap	Unix like, Win 9x/Nt/2000

Interfaccia

PORTSCAN	Grafica e riga di comando.
NESSUS	Server:riga di comando Client: grafica.
NMAP	Riga di comando.
SAINT	Grafica e riga di comando.

Architettura

PORTSCAN	Programma indipendente.
NESSUS	Client / server.
NMAP	Programma indipendente.
SAINT	Client / server.

Distribuzione

PORTSCAN	Freeware / shareware.
NESSUS	Freeware.
NMAP	Freeware.
SAINT	Freeware.

Ambito di utilizzo

PORTSCAN	Attingere soltanto informazioni su un computer.
NESSUS	Difesa infatti attacca solo per verificare la robustezza ed eventuali debolezze.
NMAP	Attacco (anche in modo anonimo).
SAINT	Difesa infatti attacca solo per verificare la robustezza ed eventuali debolezze.

Tipi di scansioni

	NMAP	NESSUS	SAINT	PORTSCAN
TCP CONNECT scan	X	X	X	X
TCP SYN	X	X	X	X
TCP FIN	X	X	X	X
TCP XMAS TREE	X	X	X	X
TCP NULL SCAN	X	X	X	X
UDP SCAN	X	X	X	

Servizi offerti

PORTSCAN	Permette di ottenere informazioni sulle porte listando *porta, servizio, descrizione*.
NESSUN	Verifica la presenza di bug relativi alla sicurezza (ad ex intrusione via ftp anonimo e propone soluzioni) ed è fortemente dipendente da plug-in.
NMAP	Oltre ad ottenere informazioni sulle porte listando *porta, stato, servizio* esegue il riconoscimento del sistema operativo e supporta IP DEVOY per nascondersi all'attaccato.
SAINT	Attinge informazioni circa le porte, riconosce il sistema operativo, logger di users e password, verifica la presenza di bug e propone soluzioni. Effettua una visita in profondità della topologia della subnet interrogato.

Banco di prova

I software sono stati provati sui seguenti sistemi operativi

Portscan	Windows 2000
Nmap	Linux Suse 6.0
Nessus	Linux Suse 6.0
Saint	Linux Suse 6.0

Spazio su disco

Saint	3Mb Base + 70Mb Plug in
Portscan	3Mb
Nessus	2Mb Base + ?Mb Plug in
Nmap	4Mb

Portscan

Esegue il controllo delle porte  all'interno di una rete di macchine comunicanti in TCP/IP.

Come scrivere un semplicissimo Portscan in C partendo da zero

Quello che si illustra è del più semplice e meno trasparente portscanner.

E' molto utile a scopo didattico.

Ha come riferimento il C/ANSI e le librerie UNIX.

Le fasi sono:

- Aprire un po di files in C
- Portscan in C

```
#include <stdio.h>
Main()
{
 int n;
 FILE *fp;
 char nome_file[10];
 for (n=1;n<1025;n++)
   {
    sprintf (nome_file,"file_%d",n);
    if ( (fp=fopen(nome_file, "r")) != NULL )
      printf("\nfile %s trovato!",nome_file);
   }
}
```


- al posto di "fopen()" si usa "connect()“
- si usa "socket()“ per creare il descrittore.

I tipi di dati utilizzati sono:

```
int sock;

“ un particolare tipo che contiene le informazioni sul servizio al quale
ci stiamo connettendo ”:
struct sockaddr_in {
short int          sin_family;  // info sul protocollo da utilizzare
struct             in_addr;     // contenente l'indirizzo del server
unsigned short int sin_port;    // porta alla quale si tenta la
connessione
unsigned char sin_zero[8]; // per motivi di compatibilita'
}
```


I tipi di dati utilizzati sono:

```
int sock;

“ un particolare tipo che contiene le informazioni sul servizio al quale
ci stiamo connettendo ”:
struct sockaddr_in {
short int          sin_family;  // info sul protocollo da utilizzare
struct             in_addr;     // contenente l'indirizzo del server
unsigned short int sin_port;    // porta alla quale si tenta la
connessione
unsigned char sin_zero[8]; // per motivi di compatibilita'
}
```

```
“ a sua volta la struttura in_addr e' cosi' definita ”:
struct in_addr {
unsigned long int  s_addr;      // ip del server
}
```


int socket(int domain, int type, int protocol)

la famiglia di protocolli da utilizzare, ovvero IP.

int socket(int domain, int type, int protocol)

la "semantica" per costruire pacchetti

int socket(int domain, int type, int protocol)

altre eventuali specificazioni

Socket per comunicare con protocollo IP tramite TCP

```
Int sock;
Sock=socket(AF_INET, SOCK_STREAM, 0);
```



```
int connect(int sockfd,
            const struct sockaddr *serv_addr,
            socklen_t addrlen);
```

Il socket precedentemente creato


```
int connect(int sockfd,
            const struct sockaddr *serv_addr,
            socklen_t addrlen);
```

la struttura contenente le informazioni necessarie alla connessione.


```
int connect(int sockfd,
            const struct sockaddr *serv_addr,
            socklen_t addrlen);
```

La lunghezza della struttura passata.

Per risolvere il problema del byte order useremo funzioni apposite come:

unsigned long int inet_addr(const char *cp);

Che da un ip scritto nella notazione standard(“12.0.0.1”) calcola il corssipondente unsigned log già nel Network Byte Order.

Per risolvere il problema del byte order useremo funzioni apposite come:

```
char *inet_ntoa(struct in_addr in);
```

Dalla struttura sockaddr_in.in_addr ci restituisce un puntatore ad una stringa di caratteri della notazione standard.

Per risolvere il problema del byte order useremo funzioni apposite come:

```
unsigned short int htons(unsigned short int hostshort);
```

htons, significa Host_TO_Netowork_Shorts, che useremo per convertire il numero della porta dall' Host Byte Order (23, 80, 31337) al Network Byte Order (5888, 20480, 27002).

Implementazione

```
#include <stdio.h>
#include <sys/socket.h>
#include <netinet/in.h>

main(int argn, char **argv)
{
struct sockaddr_in indirizzo;
 int porta, sock, z;
// ora inizia subito il ciclo che fara' scannare tutte le porte comprese
// tra le due specificate nella command line
for (porta=atoi(argv[2]); porta<=atoi(argv[3]); porta++)
    {
// creo il socket con una verifica di eventuali errori
    if ((sock=socket(AF_INET, SOCK_STREAM, 0))==-1) perror("Socket:");
    else {
// ... comincio a riempire i campi della struttura sockaddr_in
        Indirizzo.sin_family=AF_INET;
        indirizzo.sin_addr.s_addr=inet_addr(argv[1]);
        indirizzo.sin_port=htons(porta);
// cerco di instaurare la connessione
        if (z=connect(sock,(struct sockaddr*) &indirizzo, sizeof(indirizzo))==0)
            printf ("\nfound port %d open!", porta);
        }
    close (sock);
    }
}
```

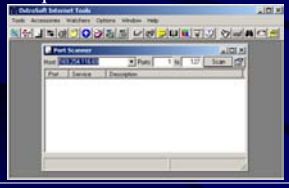

TCP Portscan for Java 1.0
Sistema op.:Win9x,WinNt
Hoppa Portscanner 2.0
Sistema op.:WinNt
Ostrosoft Portscan 2.0
Sistema op.:Win9x
TCP/IP Portscanner 1.7
Sistema op.:OS2

Ostrosoft Internet Tools

Il Portscanner visionato e analizzato e quello della Ostrosoft e si è usato il pacchetto

Ostrosoft Internet Tools (0.I.T.) 5.3

che lo integra con la sua ultima release. Esso è scritto in Visual Basic ed è primariamente rivolto all'analisi della sicurezza e alle implicazioni dei risultati.



O.I.T. integra un insieme di utility di informazioni di rete:

- Domain Scanner
- Port Scanner
- Scan Wizard
- Ping
- Traceroute
- Netstat - Host Resolver
- Network info
- Local Host Info

- Scan Wizard - analizza host, domini o intervalli IP per porte specificate. Permette inoltre di specificare un elenco di host, un elenco di porte, salvare gli scenari, etc.
- Clear History Menu, permette di cancellare la storia dell'applicazione
- Supporto per linea di comando
- File di output per ciascuna utilità

Sono supportate le seguenti linee di comando:

Scan Wizard	scanwiz scenario:[scenario name]
Port Scanner	portscan host:[host name or IP address] from:[port number] to:[port number]
Domain Scanner	domscan domain:[domain name] port:[port number]
Ping	ping host:[host name or IP address]
Traceroute	tracert host:[host name or IP address]
NS Lookup	nslookup dns:[server name] type:[*/A/NS/CNAME/SOA/MX] query:[query string]
Host Resolver	resolve host:[host name] ip:[IP address]
Netstat	netstat
Network Info	netinfo host:[host name]
Local Info	localhost
Finger	finger email:[email address]
HTML Viewer	htmlview url:[url]
Ph	ph server:[server name] query:[query string]
Simple Services	sns server:[server name] service:[echo/discard/daytime/qotd/chargen]
TCP Client	tcp host:[host name or IP address] port:[port number]
Whois	whois server:[server name] query:[query string]
Host Watcher	Whost
Service Watcher	wservice
Mail Watcher	wmail
HTML Watcher	whtml
Service Watcher	wconn

La linea di comando può includere anche i seguenti switches :

/r – run

/x – exit when finished

/h – hide

Esempio:

c:\progra~1\ostros~1\ostronet portscan host:localhost from:1 to:30 /r /h /x

La sintassi sopradescritta analizza localhost per porte attive nel range 1-30 in background e quando ha finito chiude Ostrosoft Internet Tools.

La legalità di Ostrosoft Portscan è legata alla legalità degli scanner in generale e questa è soggetta a dibattiti.

La legalità di Ostrosoft Portscan è legata alla legalità degli scanner in generale e questa è soggetta a dibattiti

Nessuna legge è stata scritta all'indirizzo degli scanner

Simulazione di Ostrosoft PortScan

Portscan scandisce l'host oppure un indirizzo IP per individuare i servizi erogati in un particolare port range.

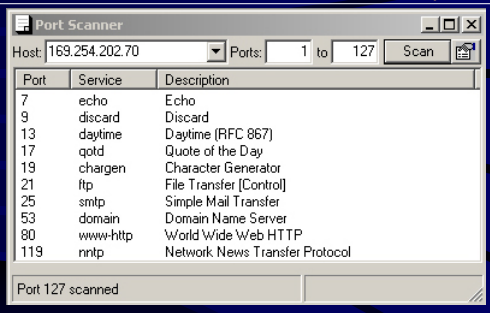
Il programma prende in input:

1) Il nome dell'host o l'indirizzo IP ad esso associato

2) L'intervallo delle porte da controllare

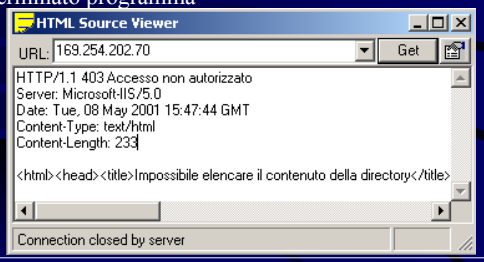
Non resta quindi che fare click sul bottone Scan

per avviare la scansione



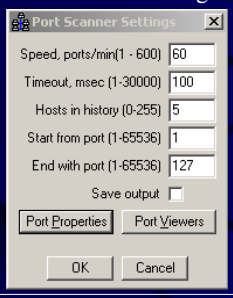
Il Programma da in output una lista delle porte aperte, il servizio ad esse associato con relativa descrizione

Dopo che la scansione è completata mediante un doppio click su ognuna delle porte presenti nell'elenco di output si può prender visione se vi è associato un determinato programma

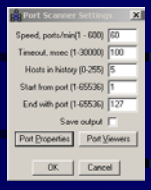


Ciccando sul bottone  si aprirà la finestra

"Port Scanner Setting"



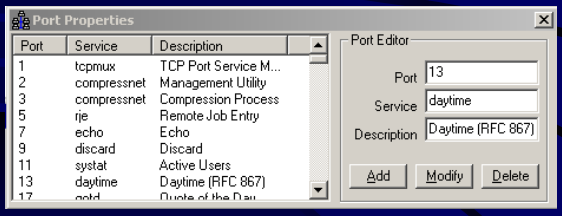
Dove l'utente può configurare:



- La velocità della scansione
- Il tempo massimo di attesa di ogni socket
- Il numero di host da tenere nella storia (della scansione)
- L'intervallo di porte da controllare per default
- Se scegliere di salvare l'output

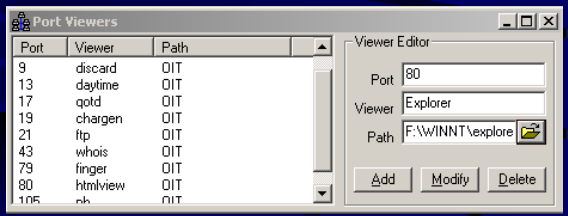
Cliccando sul bottone Port Properties

si aprirà una finestra dove è possibile editare informazioni circa le porte



E per concludere cliccando sul bottone Port Viewers

si aprirà una finestra dove è possibile editare informazioni circa le informazioni di visualizzazione

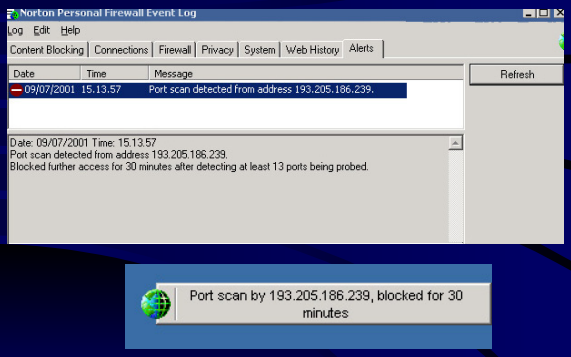


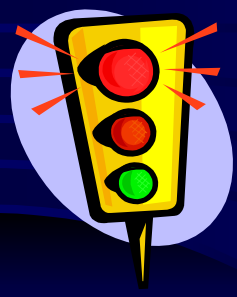
Gli strumenti in grado di proteggere il sistema da qualsiasi intrusione, comprese le scansioni sono i Firewall i quali hanno la facoltà di rilevare e bloccare scansioni

Sulla macchina obiettivo è stato installato e ben configurato il

Norton Personal Firewall 2001 Trial

il quale ha rilevato e bloccato l'attacco

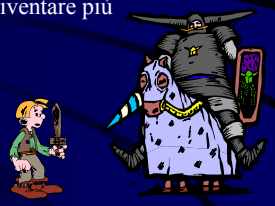




Fate attenzione quando ponete macchine sulla rete

Diminuite al minimo la possibilità di fare accesso alle vostre macchine da postazioni remote, al di fuori della vostra sottorete di indirizzi TCP/IP

Una semplice scansione delle porte aperte, tramite un portscanner, non è gran cosa; ma se porta poi ad un successivo attacco, allora il problema potrebbe diventare più importante



Nmap

COSA FA :

- Controlla le porte attive
- Rileva il Sistema Operativo

Caratteristiche di Nmap

- Scansione parallela delle porte
- Specificazione flessibile delle porte
- Specificazione flessibile dell' host
- Rilevazione degli host inattivi

Come scandire un sistema Unix

- Come usare Nmap per testare la rete
- Quali tecniche sono usate

Via con Nmap

- Dato l'indirizzo ip della macchina obiettivo
- Rileviamo i servizi attivi nella rete in esame
- Rileviamo i sistemi attivi e collegati alla rete

Via con Nmap

- Per i servizi usiamo la tecnica del ping sweep

operazione che verifica quanti e quali host sono collegati utilizzando l'invio di due pacchetti icmp:

- ECHO replay tipo 0
- ECHO request tipo 8

Via con Nmap

- Per i servizi usiamo la tecnica del ping sweep se:

i router o firewall della rete permettono il passaggio di msg icmp

- Altrimenti:

ci serviamo di un servizio interno erogato dalla macchina

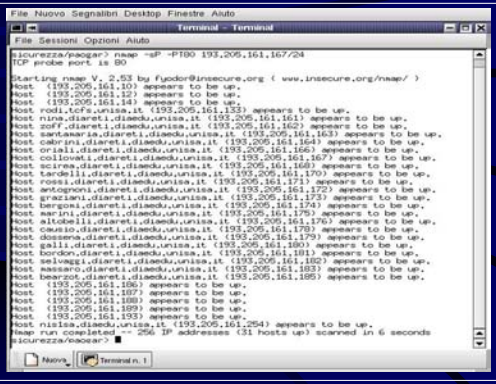
TCP Ping scan

- Serve per vedere se un host è up

- Usato per sapere quali host in una rete sono online con l'uso della opzione –sP
- Realizzato spedendo richieste ICMP echo a ciascun IP nella rete specificata
- Gli host che rispondono sono attivi

http://www.microsoft.com

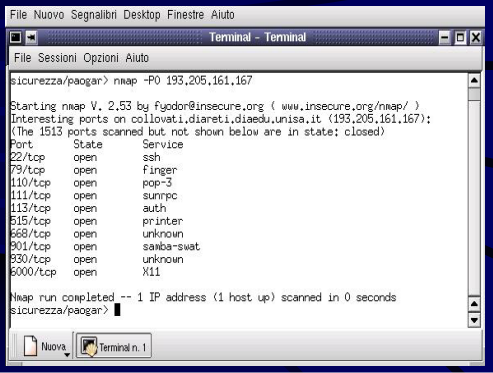
- Usato il ping TCP con l'ulteriore opzione **-PT**
- Gli host attivi dovrebbero rispondere con un RST



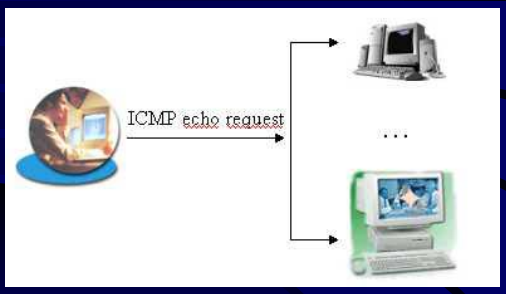
Abbiamo determinato i sistemi attivi anche dietro un firewall in grado di bloccare dei pacchetti icp

- **-P0** o **-PT80** non pingano gli host del tutto prima di sottoporli a scannimg

Usati quando facciamo portscan a http://www.microsoft.com



- **-PI**



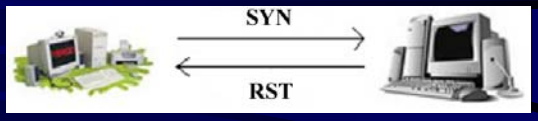

```
File Nuovo Segnalibri Desktop Finestre Aiuto
Terminal - Terminal
File Sessioni Opzioni Aiuto
sicurezza/paogar> nmap -PI 193.205.161.167

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Warning:  You are not root -- using TCP pingscan rather than ICMP
Interesting ports on collovati.diareti.diaedu.unisa.it (193.205.161.167):
(The 1513 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
79/tcp     open        finger
110/tcp    open        pop-3
111/tcp    open        sunrpc
113/tcp    open        auth
515/tcp    open        printer
668/tcp    open        unknown
901/tcp    open        samba-swat
930/tcp    open        unknown
6000/tcp   open        X11

Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
sicurezza/paogar>
Nuova
Terminal n. 1
```


- Usato il Tcp connect() scan con l'opzione **–sT**
- Se la porta è in ascolto
  connect() ha successo
  altrimenti
  la porta non è raggiungibile

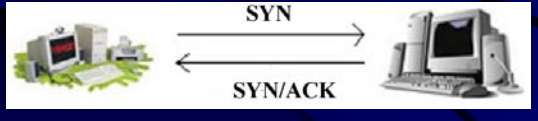
Non ha bisogno di nessun privilegio speciale


```
File Nuovo Segnalibri Desktop Finestre Aiuto
Terminal - Terminal
File Sessioni Opzioni Aiuto
sicurezza/paogar> nmap -sT 193.205.161.167

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on collovati.diareti.diaedu.unisa.it (193.205.161.167):
(The 1513 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
79/tcp     open        finger
110/tcp    open        pop-3
111/tcp    open        sunrpc
113/tcp    open        auth
515/tcp    open        printer
668/tcp    open        unknown
901/tcp    open        samba-swat
930/tcp    open        unknown
6000/tcp   open        X11

Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
sicurezza/paogar>
Nuova
Terminal n. 1
```


- **-PS**



oppure

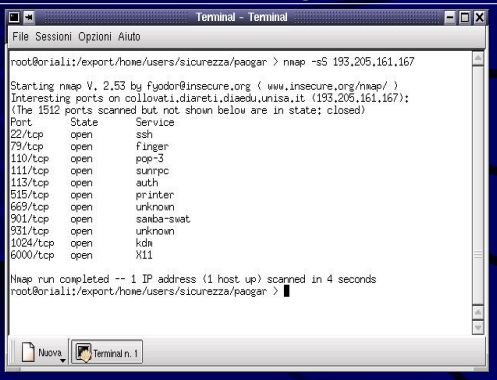



```
Terminal - Terminal
File Sessioni Opzioni Aiuto
sicurezza/paogar> nmap -PS 193.205.161.167

Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
Interesting ports on collovati.diareti.diaedu.unisa.it (193.205.161.167):
(The 1512 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
79/tcp     open        finger
110/tcp    open        pop-3
111/tcp    open        sunrpc
113/tcp    open        auth
515/tcp    open        printer
668/tcp    open        unknown
901/tcp    open        samba-swat
931/tcp    open        unknown
1024/tcp   open        kdm
6000/tcp   open        X11

Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
sicurezza/paogar>
Nuova
Terminal n. 1
```

Come Nmap realizza il Port Scanning

- Controlliamo i servizi attivi di una ipotetica macchina 193.205.161.167 nascosta dietro a un firewall

Come Nmap realizza il Port Scanning



Come Nmap realizza il Port Scanning

- Usato il tcp syn con l'opzione –sS per connessioni non complete ma capace di rilevare il servizio attivo su una determinata porta

Quanto è sicuro il SYN scan?

Può essere rilevato, e il nostro ip reale può essere così svelato

Effettuare lo scanning da più indirizzi ip contemporaneamente

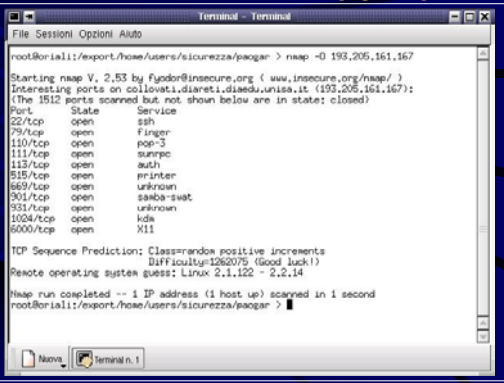
Come interviene Nmap?

- Permette di specificare uno o piu' indirizzi "falsi" come ip sorgente: oltre a quello reale invia pacchetti "devoy" con l'ip sorgente ritoccato

Come realizzare ulteriori tecniche di scanning

- - sF per la modalità di scan Stealth FIN
- - sX per la modalità di scan Xmas Tree
- - sN per la modalità Null scan

- Riusciamo a sapere anche di quale Sistema Operativo dispone il nostro bersaglio
- Probabilmente anche un indicazione sul Kernel utilizzato



Secondo Nmap si tratta di Linux con un possibile Kernel 2.2

- Non fornisce una shell grafica quindi le operazioni vanno inserite come linee di comando
- Non necessariamente testa la presenza di tutti gli host attivi sulla rete ma solo quelli di cui siamo interessati

- Individuate le porte aperte non necessariamente effettua lo scanning su tutte prima di attaccare ma può scegliere una porta specifica come bersaglio

E' più veloce la computazione delle operazioni eseguite dall' amministartore di sistema o da chiunque usi questo programma

- Il codice di Nmap è scritto in C un linguaggio comunque evoluto ed anche facilmente modificabile
- Portabilità

Nessus

Cos'è Nessus?

- Un Port Scanner

- Un Server (solo Unix-like)
- Un Client (anche Win32)
- Molti Servizi (test di sicurezza)

Come è Nessus

- Gratuito
- Open-Source
- Aggiornato (Plug-in)
- Pignolo
- Scaltro

Utenti Nessus

- Il sysad è l'utente del server
- Gli utenti del sistema "possono" essere utenti di Nessus client
- Il sysad stabilisce nome utente e password per ogni utente
- Ogni utente sarà provvisto di una coppia di chiavi pubblica e privata
- L'accesso degli utenti avviene in modo autenticato
- Le comunicazioni avvengono in modo cifrato
- Ciascun utente può avere delle restrizioni sul numero di macchine che può esaminare

Architettura Plug in

- Ciascun Test di Sicurezza è scritto come un plug-in esterno
- Giornalmente, chiunque può segnalare un nuovo bug
- Esiste una lista di plug-in ancora da scrivere
- Qualunque utente registrato può offrirsi volontario per scrivere un plug-in
- E' scritto in C o in Nessus Attack Script Language (NASL)
- Ciascun plug-in sarà testato dall'autore di Nessus

Cosa offre Nessus

- SICUREZZA
  - Le comunicazioni tra client e server avvengono in modo cifrato
  - Gli accessi avvengono in modo autenticato
- AFFIDABILITA'
  - I test di sicurezza sono in continuo aggiornamento
  - Tutti i test sono garantiti dall'autore

Nessus all'opera: Creazione utenti

Sul server..

Nessus all'opera: Creazione utenti

Terminal – Terminal

File Sessions Options Help

```
altafini:~ # nessusd -L
                  buio@193.205.161.192 - user key
                gianni@193.205.161.192 - user key
altafini:~ #
```

New | Terminal No 1

Nessus all'opera: Creazione utenti

Terminal – Terminal

File Sessions Options Help

```
altafini:~ # nessusd -L
                  buio@193.205.161.192 - user key
                gianni@193.205.161.192 - user key
altafini:~ # nessusd -P maja,maja
altafini:~
```

New | Terminal No 1

Nessus all'opera: Creazione utenti

Terminal – Terminal

File Sessions Options Help

```
altafini:~ # nessusd -L
                  buio@193.205.161.192 - user key
                gianni@193.205.161.192 - user key
altafini:~ # nessusd -P maja,maja
altafini:~ # nessusd -L
                  buio@193.205.161.192 - user key
                gianni@193.205.161.192 - user key
                                  maja - user password
altafini:~
```

New | Terminal No 1

Nessus all'opera: Creazione utenti

Sul client…

generazione numeri primi

Nessus all'opera: Creazione utenti

Nessus

To protect your private key just generated, enter your personal pass phrase, now. Keep that pass phrase secret. And each time when you restart nessus, re-enter that pass phrase when you are asked, for. This prevents anybody else from logging in to the nessus server using your account.

The drawback of a pass phrase is that it will prevent you from being able to use nessus(1) in a cron job or in a quiet script.
If you do not want to use a pass phrase, enter a blank one.

To change or remove the pass phrase, later on read in the manual page nessus(1) about the -C option.

Passphrase :

Passphrase (again) :

Cancel | Ok

Nessus all'opera: Creazione utenti

Nessus Setup

Nessusd host | Plugins | Prefs. | Scan options | Target selection | User | Credits

New session setup

Nessusd Host : 193.205.161.191

Port : 3001

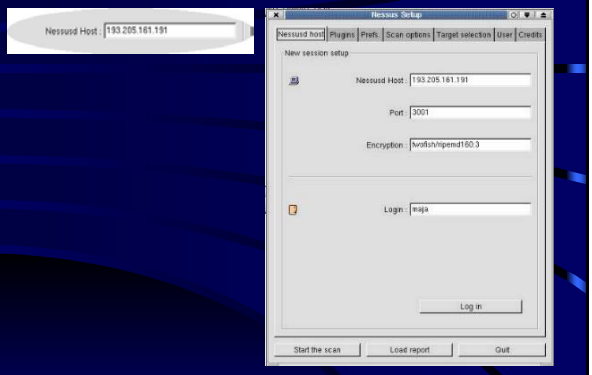
Encryption : lwo6ish/ripemd160.3

Login : maja

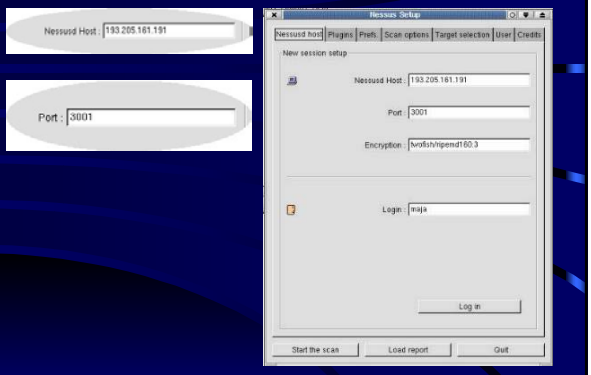
Log in

Start the scan | Load report | Quit

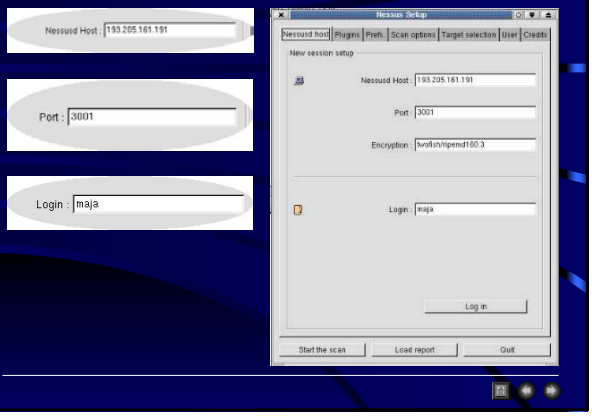
Nessus all'opera: Creazione utenti



Nessus all'opera: Creazione utenti



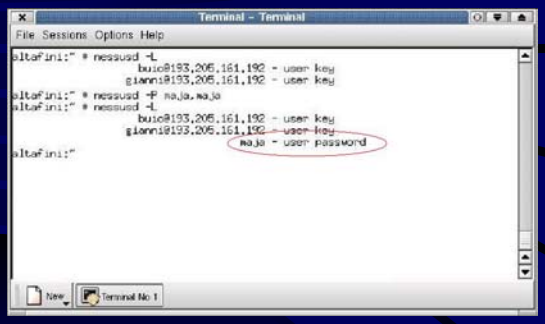
Nessus all'opera: Creazione utenti

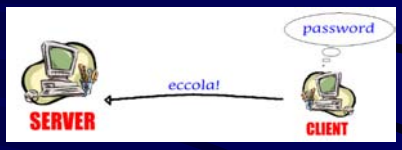


Nessus all'opera: Creazione utenti

Nessus all'opera: Creazione utenti

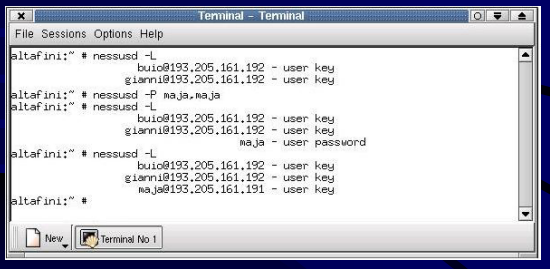
Nessus all'opera: Creazione utenti





- Il client è connesso e può effettuare i test!

- Il key ring del Server

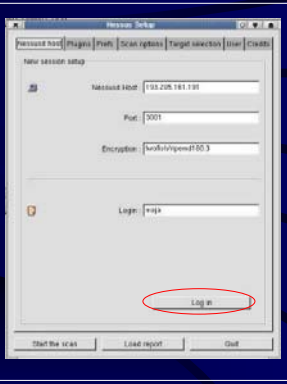


Nessus all'opera: i key ring

- Il key ring del Client

Nessus all'opera: connessioni successive

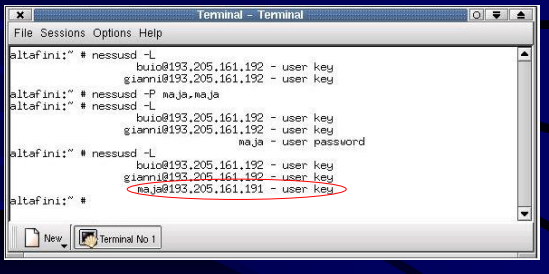
Nessus all'opera: Testing!



Nessus all'opera: connessioni successive

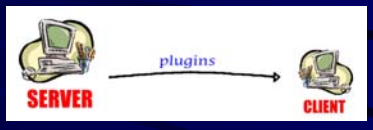
Nessus all'opera: Creazione utenti

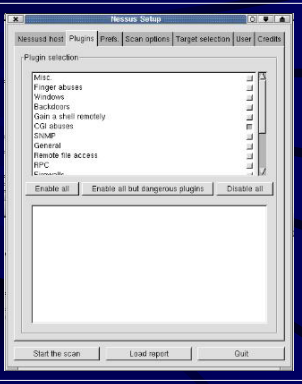
Nessus all'opera: i key ring

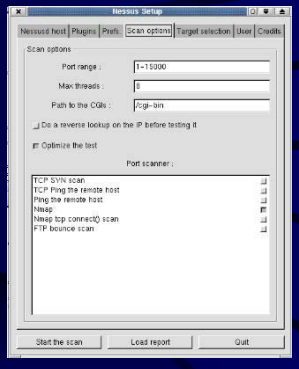


- Il client è connesso e può effettuare i test!

Dopo l'autenticazione..

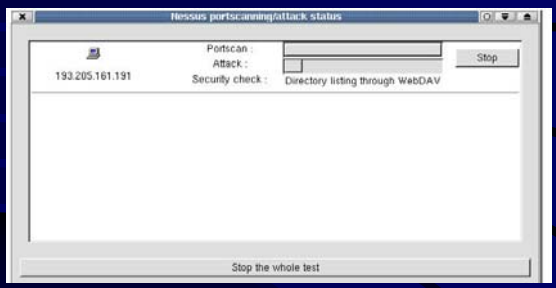


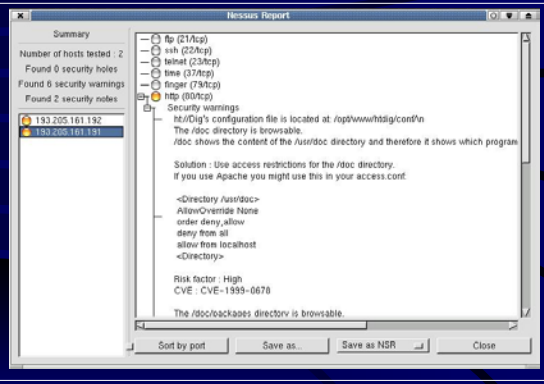




- Utenti più esperti e con esigenze particolari possono usare le specifiche più avanzate

Il client invia al server la richiesta





Saint

Security
Administrator 's
Integrated
Network
Tool

- Tool per l'analisi delle vulnerabilità della rete
- Versione aggiornata di SATAN

Ultima versione SAINT 3.3.4 (7/03/2001) + Plug-in:

- SAINTwriter - migliore analisi grafica dei dati
- SAINTexpress – updates automatico delle vulnerabilità

Come è Saint

- Open Source
- Simile a ISS, CyberCop, Nessus, SATAN …

Standardizzazioni

- Compatibile con CVE (Common Vulnerabilities and Exposures)
- Certificato SANS (System Administrator Network Security)

Cosa fa?

- Individua i Target Hosts (ping/TCP scan)
- Indentifica i servizi (TCP scan/UDP scan)
- Analizza i servizi trovati
- Individua le vulnerabilità
- Riporta la topologia della rete
  - produce una lista di host UNIX per tipo di S.O
  - Identifica i web servers
  - Identifica i routers

Cosa non fa?

- Test su altri protocolli (es. IPX)
- Eseguire qualche test sul Denial of Services
- Modalità operativa "stealth" (segreta)

Risorse sw richieste da Saint

- Compilatore C
- PERL 5.004
- Web browser
- Samba utilities (per SMB tests) – optional
- NMAP (per test del S.O) – optional

Come avviene la raccolta dei dati

Policy engine

- Controlla gli host che SAINT può sondare
- Controlla la profondità di scansione
- I Parametri sono specificati in saint.cf
- Le modifiche possono essere effettuate da linea di comando o dall' interfaccia HTML

Target Acquisition

- Specificati dall'utente (nel modulo "Data Acquisition"):
  - one host
  - subnet scan
  - range scan
  - list (qualche combinazione di host, subnet and range)
  - target file
- Generati dal modulo "Inference Engine"
- Controlla se gli host sono up o down

Data Acquisition

- Esegue un sottoinsieme delle sonde in base al livello di scansione stabilito:
  - Light – inf. Dal DNS, stabilisce il S.O. e servizi RPC
  - Normal – servizi di rete comuni: ftp, www, finger etc…
  - Heavy – analizza I servizi e controlla l'anonymous FTP
  - Heavy+ - uguale a Heavy ma non evita le porte critiche
  - Top10 – analizza le top 10 vulnerabilità specificate da SANS
  - Custom – sonde specificate dall'utente

Inference engine

- Input: dati raccolti dal modulo "Data Acquisition"
- Output:
  - Nuovi facts per il modulo "Inference Engine"
  - Nuove sonde per il modulo "Data Acquisition"
  - Nuovi Target per il modulo "Target Acquisition"
- Tutto è gestito da una base di regole
- La raccolta continua fino a che non viene trovato niente di nuovo (ricerca breadth-first)

Formato del Database

- All-hosts - tutti gli host incontrati (esplorati e non)
- Facts - dati generati dai moduli "Data Acquisition" e "Inference Engine"
- Todo - lista di tutte le probes eseguite sugli host
- Cve - le vulnerabilità che corrispondono alle top10 SANS

Formato del Database: All-Hosts

All-Hosts :

- Host name
- IP address
- Livello di prossimità dall'host origine
- Attack level host
- Espansione di sottorete (1 = yes, 0 = no)
- Tempo impiegato per lo scan

Esempio All-Hosts

193.205.161.19|193.205.161.19|0|0|0|

altafini.diareti.diaedu.unisa.it|193.205.161.191|0|3|1|994674968

tancredi.diareti.diaedu.unisa.it|193.205.161.192|0|3|1|994674970

193.205.161.128/|||||994083702

Formato del Database: Facts

- Target - nome dell'host a cui si fa riferimento
- Service - nome base di tool o servzi rilevati dalle probes
- Status - raggiungimento host
- Severity - gravità delle vulnerabilità
- Trust – fiducia nel target
- Trusted – chi si fida di chi (user@host)
- Canonical Service Output
  - for non-vulnerability records, the reformatted version of the network service
  - for vulnerability records, the name of the tutorial
- Text – informazioni addizzionali

Esempio Facts

- altafini.diareti.diaedu.unisa.it|telnet|a||||Welcome to SuSE Linux 7.1 (i386) - Kernel 2.2.18 (2).\r\nUSER saint\r\nNICK q\r\nQUIT\r\n\r\n\r\n\r\naltafini login: |offers telnet
- tancredi.diareti.diaedu.unisa.it FTP server (Version 6.5/OpenBSD, linux port 0.3.2) ready.\r\n331 Password required for saint.\r\n500 'NICK q': command not understood.\r\n221 Goodbye.\r\n|offers ftp

Formato del Database: Todo

Todo:

- Host name
- Tool da eseguire dopo
- Argomenti per i tool

Esempio Todo

altafini.diareti.diaedu.unisa.it|ftp.saint|
tancredi.diareti.diaedu.unisa.it|pop3.sara|
tancredi.diareti.diaedu.unisa.it|http.saint|http
altafini.diareti.diaedu.unisa.it|login.saint|-r -u wank -p wank telnet
altafini.diareti.diaedu.unisa.it|xhost.saint|-d altafini.diareti.diaedu.unisa.it:0
altafini.diareti.diaedu.unisa.it|ostype.saint|
altafini.diareti.diaedu.unisa.it|http.saint|1148
tancredi.diareti.diaedu.unisa.it|tcpscan.saint
12754,15104,16660,20432,27665,1-9999|

Formato del Database

CVE:

- Top 10 flag: Se la vulnerabilità è o meno sulla lista Top 10 di SANS
- CVE name(s): Il nome CVE o i nomi corrispondenti alle vulnerabilità
- Vulnerability Text: campo Text del database Facts

yes||Exports /tcfsexport to everyone

yes|1999-0018 1999-0019 1999-0210 1999-0493 2000-0666|rpc.statd is enabled and may be vulnerable

yes|1999-0002|mountd may be vulnerable

no|2000-0389 2000-0390 2000-0391|Is your Kerberos secure?

- Interfaccia grafica in HTML (default)
  - Data Managment
  - Target Selection
  - Config Managment
  - Data Analisys
- Linea di comando
  - ./saint [options] [target1] [target2] …
- In remoto
  - ./saint -r

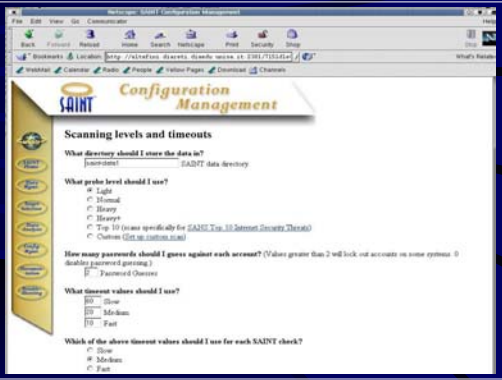
Modulo Magic cookie generator

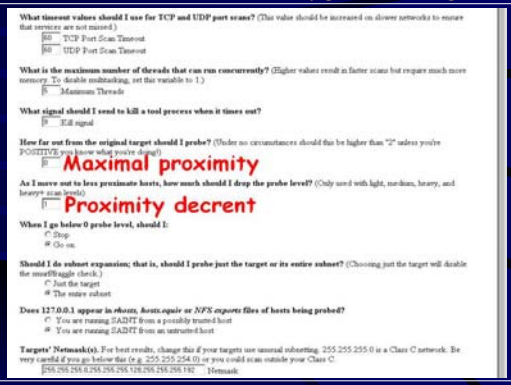
- Avvia un daemon http di SAINT
  - Sottoinsieme del daemon httpd
- Genera una stringa pseudocasuale di 32 byte
  - Inviata dal web browser a Saint in ogni comando
  - Scopo: differenziazione di tutti i processi Saint.
  - Viene generata una nuova chiave per ogni sessione
- Esamina tutti i dati precedentemente raccolti
  - Per default in $$saint_data

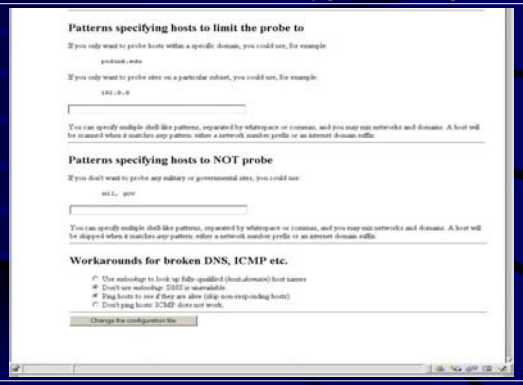
- Aprire un database
- Crearne uno nuovo
- Fondere database
- Default saint_data

Si può specificare :

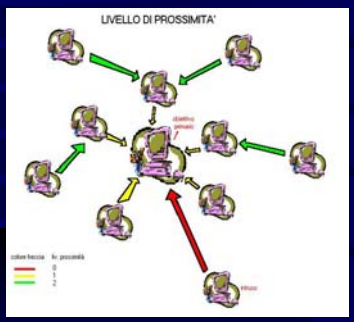
- Un host (nome o IP)
- Lista di indirizzi IP
- Un range di Ind. IP
- Un'intera sottorete
- Target_file
- Livello di scansione
- Supporto firewall







- $Maximal proximity
- $ Extreme



- Massimo livello di prossimità
- Evitare valore troppo alto (2)
- Variabile Proximity decrent

- Governa i test pericolosi
- Livello di scansione Heavy+

- Vulnerabilità
- Inf. Host (topologia della rete)
- Trust

Metodi per l'esame dei risultati

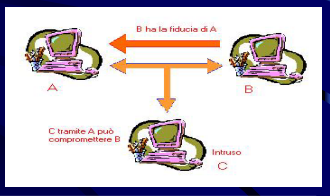
- By approximate danger level
  - Ordina tutti i problemi in base al Severity Levels
- By type of vulnerability
  - Mostra tutti i tipi di vulnerabilità trovati più una lista degli host corrispondenti
- By vulnerability count
  - Visualizza tutti gli host che hanno molti problemi

- Critical Problems (Red)
- Areas of Concern (Yellow)
- Potential Problems (Brown)
- Services (Green)
- Other Information (Black)
- Top10

- Categorie di informazioni :
  - Tipo di servizio (Anonimous FTP, www ,etc …)
  - Tipo del sistema (stabilito da Nmap se disponibile)
  - Internet domain (Host divisi per domini DNS)
  - Sottorete (per Saint 256 ind. IP adiacenti)
  - Host name (una query al database per le inf sull' host)

Come si ottiene la fiducia?

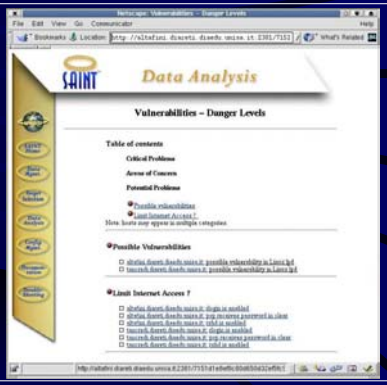
- Dai file".rhost" e "host.equiv"
- "Window Server"
- Transitivamente



SAINT data collection. Data collection in progress...

07/09/01-12:32:32 bin/timeout 120 bin/get_targets tancredi.diareti.diaedu.unisa.it
07/09/01-12:33:40 bin/timeout 20 bin/ostype.saint tancredi.diareti.diaedu.unisa.it .PLUS
07/09/01-12:33:40 bin/timeout 20 bin/rpc.saint tancredi.diareti.diaedu.unisa.it .PLUS
07/09/01-12:33:40 bin/timeout 20 bin/finger.saint tancredi.diareti.diaedu.unisa.it .PLUS
07/09/01-12:33:40 bin/timeout 60 bin/tcpscan.saint 12754,15104,16660,20432,27665,1-9999 altafini.diareti.diaedu.unisa.it .PLUS
07/09/01-12:33:40 bin/timeout 20 bin/ostype.saint altafini.diareti.diaedu.unisa.it .PLUS
07/09/01-12:33:40 bin/timeout 60 bin/udpscan.saint 19,53,69,111,137-139,161-162,177,8999,1-18,20-52,54-68,70-110,112-136,140-160,163-176,178-1760,1763-2050,32767-33500 altafini.diareti.diaedu.unisa.it .PLUS
07/09/01-12:33:42 bin/timeout 20 bin/dns.saint tancredi.diareti.diaedu.unisa.it .PLUS
07/09/01-12:33:43 bin/timeout 20 bin/finger.saint altafini.diareti.diaedu.unisa.it .PLUS
07/09/01-12:33:51 bin/timeout 20 bin/ddos.saint tancredi.diareti.diaedu.unisa.it .PLUS
07/09/01-12:33:57 bin/timeout 20 bin/rpc.saint altafini.diareti.diaedu.unisa.it .PLUS
07/09/01-12:33:57 bin/timeout 20 bin/ddos.saint altafini.diareti.diaedu.unisa.it .PLUS
07/09/01-12:34:00 bin/timeout 60 bin/tcpscan.saint 12754,15104,16660,20432,27665,1-9999 tancredi.diareti.diaedu.unisa.it .PLUS
07/09/01-12:34:03 bin/timeout 20 bin/dns.saint altafini.diareti.diaedu.unisa.it .PLUS
07/09/01-12:36:10 bin/timeout 20 bin/mountd.sara tancredi.diareti.diaedu.unisa.it .PLUS
07/09/01-12:36:10 bin/timeout 20 bin/printer saint tancredi.diareti.diaedu.unisa.it .PLUS
07/09/01-12:36:10 bin/timeout 20 bin/login.saint -r -u wank -p wank telnet tancredi.diareti.diaedu.unisa.it .PLUS
07/09/01-12:36:32 SAINT run completed
Data collection completed (2 host(s) visited).

Data collection



Saint da linea di comando

- /saint [options] [target1] [target2] …
  - hosts name
  - IP addres
  - IP subnet
  - IP addres ranges
- ./saint – H per vedere tutte le options

Modalità remota

Caratteristiche di gestione:

- Host based access control
- User autentication
- Server Port

Come usare Saint in remoto

1. Connessione al server (riga di comando)
   - IP -h
   - Porta -p

Come usare Saint in remoto

1. Connessione al server (file config/saint.cf)
   - IP $allow_host
   - Porta $server_port

Come usare Saint in remoto

2. Avviare saint in remoto (./saint –r)
3. Settare le passwd per admin e altri utenti
4. Dal browser scrivere http://host.domain:port
5. Login
6. Logout

Come creare un nuovo utente

- La lista di utenti è il file config/password
- Creare un utente = Aggiungere una riga

<login name>:<password cifrata>:<user ID>:<commenti>

Conclusioni

- potente e versatile
- semplice ma delicato
- un'architettura estendibile
- semplice aggiungere nuove funzionalità
- facile da installare e configurare
- ingordo di memoria

Bibliografia

La realizzazione di questa presentazione è stata fatta attingendo informazioni dai seguenti siti web :

http://www.ostrosoft.com
http://www.nessus.org
http://www.insecure.org/nmap
http://www.saint.com

Verifica robustezza dei servizi di rete: gli scanner

Università degli studi di Salerno
Facoltà di Scienze MM FF NN
Corso di laurea in Informatica
Dipartimento di Informatica ed applicazioni

Progetto di

Sistemi di elaborazione dell'informazione
(Sicurezza su reti)
Prof. Alfredo De Santis
Anno accademico 2000 / 2001

Autori del progetto
Alessandro Ursomanno matr.: 056 / 001050
Daniele Mallozzi matr.: 056 / 100063
Giovanni Vernacchio matr.: 056 / 100306
Paola Gargiulo matr.: 056 / 00460