Vulnerabilita'

 1

- Le vulnerabilita' note sono in numero enorme (vedere archivi mantenuti da SecurityFocus o CVE, ad esempio)
- Relative sia a servizi di rete tradizionali (es. telnet, ftp) che ad applicazioni specifiche (es. MS IIS, Oracle)
- Tutte dipendono dalla specifica implementazione (produttore, versione)
- Esistono patch rilasciate dai produttori per tutte le vulnerabilita' (o quasi)

Riferimento interessante: **SANS Institute**

The twenty most critical Internet security vulnerabilities

Versione 3.1, 7 Ottobre 2002

`www.sans.org/top20.html`

 2

Analizziamo alcune delle vulnerabilita' riportate dal SANS.

Quelle riportate dal SANS sono le vulnerabilita' ritenute prioritarie, secondo il giudizio di un gruppo di esperti, da cui proteggere i sistemi.

Sistemi Windows

- Web Server IIS;
- Microsoft SQL Server;
- Condivisioni via NETBIOS;
- Logon Anonimi (Null Session);
- Autenticazioni senza password o password deboli;
- Internet Explorer;
- Accessi Remoti ai Registri di Sistema;
- Windows Scripting Host

 3

Principali funzionalita' dei sistemi Windows colpite da gravi vulnerabilita' (fonte SANS Institute).

Una osservazione generale che puo' essere fatta rispetto l'elenco mostrato e' che vulnerabilita' aventi possibili conseguenze gravi per un sistema si ritrovano in tutte le componenti principali ed in particolare nei:

-servizi o applicazioni interattive (es. Web Server, browser, condivisione di file e directory);

-Meccanismi di autenticazione e logon;

-Database.

Sistemi Unix/Linux

- Remote Procedure Calls (RPC);
- Apache Web Server;
- Secure Shell (SSH);
- Simple Network Management Protocol (SNMP);
- Autenticazioni senza password o password deboli;
- File Transfer Protocol (FTP);
- Sendmail;
- BIND/DNS
- R-Services (rlogin, rshell, etc.)

 4

Principali funzionalita' dei sistemi Unix/Linux colpite da gravi vulnerabilita' (fonte SANS Institute).

La stessa osservazione fatta in precedenza per I sistemi Windows puo' essere ripetuta per I sistemi Unix/Linux: vulnerabilita' aventi possibili conseguenze gravi per un sistema si ritrovano in tutte le componenti principali.

-servizi o applicazioni interattive (es. Web Server, file transfer, esecuzione remota di comandi, posta elettronica);

-DNS;

-Meccanismi di gestione remota (SNMP).

Da notare che ancora, l'utilizzo di account senza password o con password facilmente individuabile e' una delle piu' gravi vulnerabilita' di un sistema.

Microsoft IIS (Web Server) e Apache Web Server

Sistemi: tutti

- **IIS**: gravi vulnerabilita' nel gestire richieste malformate, buffer overflow e bachi in applicazioni esemplificative;

- **Apache**: meno frequenti rispetto IIS ma potenzialmente gravi (es. modulo di gestione di SSL), buffer overflow.

Contromisure minime

- Utilizzare solo le funzionalita' e i moduli strettamente indispensabili.
- Installare patch ed eseguirlo come utente non privilegiato, ove possibile.

Osservazione

Qualunque web server non puo' essere considerato sicuro fintanto che non si e' garantita la sicurezza di tutte le applicazioni web presenti su di esso. Basta un semplice script CGI mal progettato per compromettere anche il web server potenzialmente piu' sicuro e meglio configurato.

 5

I web server, come regola prudenziale di gestione di una rete, sono sempre da considerarsi come macchine attaccabili e potenzialmente soggette ad intrusione. Questo indipendentemente dal tipo di software, dalla configurazione etc.

E' pertanto buona regola isolare I web server in maniera tale che un intrusore, prendendone il controllo, non possa facilmente connettersi ad altre macchine critiche della rete.

Da qui, l'uso di sottoreti dedicate, comunicazioni e protocolli ridotti al minimo indispensabile, politiche di autorizzazione stringenti per qualunque connessione tra un web server e altre macchine della rete, meccanismi di monitoraggio e di rilevazione di attacchi (analizzatori di log, intrusion detection system, etc.).

Quanto sopra pero' va interpretato come misura di sicurezza indispensabile, IN AGGIUNTA ad una buona gestione dei web server stessi. La scelta del tipo di web server puo' essere guidata dalle statistiche relative alle vulnerabilita' identificate.

Una buona configurazione (hardening) rispetto ai possibili problemi di sicurezza e' indispensabile (limitare al minimo le porte applicative disponibili, evitare l'uso di servizi altamente a rischio, non far convivere sulla stessa macchina servizi con grado di rischio molto diverso, evitare meccanismi di controllo remoto se non ben collaudati, evitare le ultime release di sistemi non maturi, etc.).

Molti script ed estensioni applicative presenti su web server sono vulnerabili

Sistemi: tutti i web server

- gli script (es. CGI) operano con gli stessi privilegi del web server
- molti, compresi quelli forniti di default come esempi, presentano vulnerabilita';
- facilmente accessibili
- installati anche su macchine che non operano come web server

Contromisure minime

- Installare patch e eseguirlo come utente non privilegiato
- Eliminare tutti gli script non necessari o di esempio, testare a fondo quelli utilizzati in produzione

 6

A questo si associa il ben piu' grave problema delle vulnerabilita' introdotte dalle applicazioni di larga diffusione e dai sistemi ad-hoc sviluppati per la specifica organizzazione.

Oggi, le vulnerabilita' applicative sono il PRIMO e piu' grave veicolo di vulnerabilita' dei sistemi.

Remote Procedure Calls (RPC)

Sistemi: tutti i sistemi Unix/Linux

- le RPC permettono ai programmi su di un computer di eseguire procedure su di un secondo computer;

- largamente usate per implementare molti servizi di rete quali: gestione remota e file system condivisi (NFS);

Contromisure minime

- Disabilitare ogni servizio RPC non indispensabile;

- Installare le patches dei produttori per ogni servizio RPC attivo;

- Impedire le connessioni alle porte RPC critiche (111, 32770-32789) da Internet o da organizzazioni esterne.

 7

Le RPC sono uno dei meccanismi di base in sistemi Unix per implementare servizi di rete (analogamente a Netbios per sistemi Windows). Sono quindi installate ed abilitate di default nella maggior parte delle configurazioni standard.

Le vulnerabilita' evidenziate nel corso degli anni hanno permesso a intrusori di acquisire diritti di root sulle macchine attaccate, oltreche' essere sfruttate per provocare Denial of Service.

Vulnerabilita' in Sistemi di Gestione di Database

Sistemi: tutti i database

-i database mantengono le informazioni piu' critiche, sensibili e di valore di una organizzazione.

-installati di solito su macchine che non operano da web server ma da questi acceduti (direttamente o meno).

Contromisure minime

- Mai permettere un accesso diretto al database via Internet, evitare quanto possibile l'accesso al database direttamente da un Web Server (via cgi, servlet, etc.)

- Utilizzare meccanismi di replica (parziale) dei dati per isolare ulteriormente i database master.

 8

Microsft SQL Server ha presentato moltissime vulnerabilita' nel corso degli anni. Problemi simili, anche se con frequenza inferiore, sono stati evidenziati da TUTTI gli altri produttori di sistemi di gestione di database (es. Oracle, Lotus Notes, etc.).

I problemi di sicurezza su database sono riconducibili a due tipologie:

-accessi da personale interno non controllati ed assegnati in maniera troppo permissiva;

-Accessi da Internet o patrner esterni.

-Nel primo caso, i problemi nascono soprattutto dal fatto che spesso sono moltissimi gli operatori interni ad una organizzazione che possono accedere direttamente ai database e spesso con privilegi superiori a quanto sarebbe necessario e sufficiente. Gli accessi pertanto andrebbero limitati al minimo sia in numero che in varieta' di dati accedibili.

Nel secondo caso invece, i problemi si riconducono spesso ad una non sufficiente progettazione dell'architettura di rete e dell'architettura dei servizi applicativi. Di regola, nessun accesso diretto a database dovrebbe essere permesso dall'esterno di una organizzazione. Non solo, nessun accesso DIRETTO a database dovrebbe essere consentito a macchine dell'organizzazione ma esposte ad accessi esterni e quindi facili obiettivi di intrusione (tipicamente Web Server).

Vulnerabilita' relative a SNMP e configurazione di default

Sistemi: tutti i sistemi e molti componenti di rete (es. Stampanti)

- SNMP usato per monitoraggio e configurazione remota dei servizi di rete, estremamente diffuso
- la password (chiamata "community string") spesso e' lasciata ai valori di default `public` o `private`
- recentemente molte vulnerabilita' sono state scoperte che possono permettere sia denial-of-service che il controllo

Contromisure minime

- eliminare il servizio ove non necessario, installare patch
- utilizzare community string non di default, ove possibile
- impedire l'accesso esterno a questi servizi (porte 161 e 162 tcp/udp)

 9

SNMP e' il protocollo di rete utilizzato da strumenti per il monitoraggio e la configurazione remota di dispositivi di rete TCP/IP. Esempi di dispositivi sono: stampanti, router, switch. Una comunicazione via SNMP consiste nello scambio di messaggi tra una postazione centrale di gestione ed il dispositivo di rete. I meccanismi con cui questi messaggi vengono gestiti e i metodi di autenticazione (in particolare della postazione di gestione rispetto i dispositivi gestiti) presentano entrambi gravi vulnerabilita'.

Rispetto il meccanismo di autenticazione implementato da SNMP (versione 1 e 2), esso viene realizzato attraverso l'uso di "COMMUNITY STRING" (stringhe di riconoscimento, del tutto paragonabili a password). La comunicazione non viene crittata. Il problema piu' grave risiede nel fatto che i valori di default di tali "community string" sono noti (spesso il valore e' la stringa "public" oppure "private"). Tali valori, quasi sempre, non vengono modificati. Inoltre, molte implementazioni prevedono account di gestione o speciali, associati a community string anch'esse note pubblicamente o facilmente reperibili. In alcuni casi, addirittura, tali community string NON possono essere modificate.

Data la natura degli strumenti che utilizzano SNMP (gestione remota, riconfigurazione di dispositivi di rete), una intrusione attraverso questo protocollo puo' facilmente avere effetti disastrosi. Sono centinaia i prodotti e i componenti che implementano SNMP, il quale viene di solito abilitato nella configurazione standard di tali sistemi e sfugge molto spesso al controllo ed al monitoraggio dei responsabili di una rete.

Molte versioni di BIND permettono intrusioni con accessi di root

Sistemi: molte versioni di UNIX e Linux

- BIND e' l'implementazione piu' diffusa del servizio DNS;
- Protezioni molto scarse, spesso attivo anche su macchine che non operano da DNS;
- Una volta compromesso, usato sia per ulteriori attacchi o per fornire false informazioni;

Contromisure minime

- Disabilitare il servizio (`named`) se non utilizzato;
- Installare patch e eseguirlo come utente non privilegiato;
- Limitare al minimo le funzionalita' (es. Zone transfer) e le informazioni (es. Version String).

 10

BIND (Berkeley Internet Name Domain) e' l'implementazione piu' diffusa per un servizio di DNS, il quale permette di convertire nomi di server logici (es. www.unimi.it) in indirizzi IP.

Cosi' come per i Web Server, la necessita' per una macchina facente funzioni di DNS di essere pubblicamente contattabile ne fa un obiettivo privilegiato per le intrusioni o i denial-of-service.

Come gia' osservato per altri casi (es. SNMP), una configurazione di default ridondante di molti sistemi e la non attenta gestione da parte di molti responsabili di rete sonola fonte primaria di problemi:

-il servizio DNS viene abilitato di default anche su macchine che non devono fornire quella funzionalita' (lasciandole pero' esposte ad attacchi);

-Patch non applicate;

-Informazioni ridondanti rispetto il minimo indispensabile presenti sui DNS pubblici;

-Mancanza di monitoraggio dell'attivita' di rete dei DNS pubblici.

Login, in particolare root/administrator, senza password o con password deboli

Sistemi: tutti

- sistemi con utenze predefinite (es. Guest) con password insicure o note a priori
- politica di gestione delle password utenti spesso insicura
- amministratori di sistema che non settano le password

Contromisure minime

- Politica di gestione delle password, controlli e corsi
- uso periodico di strumenti di "password cracking" per test (vedere Nota);
- proteggere le password in luogo e con modalita' sicure

 11

Praticamente tutti i meccanismi di interazione tra operatori e computer utilizzano password per controllarne l'accesso.

Nella maggior parte dei casi, tali password sono definite a discrezione dell'utente. I problemi nascono dal fatto che:

-la password viene impostata ad un valore estremamente semplice o addirittura viene eliminata;

-La segratezza delle password non viene mantenuta;

-I sistemi talvolta possiedono account di sistema con password predefinite (talvolta non modificabili, o modificabili in modo particolarmente complesso);

Esistono strumenti ("password scanner") per determinare le password. Utilizzati sia dagli intrusori che dai responsabili di una rete per verificare la qualita' delle password settate.

NOTA: Mai utilizzare un "password scanner", anche su sistemi sotto la propria responsabilita', senza un'autorizzazione esplicita (meglio se scritta) da parte della direzione della propria organizzazione. Sono noti molti casi di gravi problemi insorti per un uso benintenzionato ma privo di autorizzazione di password scanner.

Una Politica di Uso delle Risorse Aziendali che definisca chiaramente e in maniera rigorosa i doveri e le responsabilita' di ogni utente nell'uso delle risorse e della infrastruttura tecnologica (comprese quindi le regole di gestione delle proprie password) e' indispensabile.

Notare pero' che tutte le indagini effettuate dimostrano inequivocabilmente che NON BASTA una politica d'uso ben definita e molto stringente, anche se formalizzata dalla direzione e sottoscritta da ogni utente. Solo attraverso un ampio coinvolgimento degli utenti stessi (corsi, riunioni, spiegazioni ripetute) si sono ottenuti risultati positivi realmente significativi.

Condivisione di file e informazioni via NetBios, NULL Session

Sistemi: Windows (NetBios), UNIX/Linux (NFS), Apple (AppleShare)

- condivisione di file e directory, se configurati inappropriatamente possono permettere il controllo remoto del computer
- estremamente diffusi
- utilizzati anche per la diffusione di virus

Contromisure minime

- limitare le condivisioni solo alle informazioni strettamente necessarie
- utilizzare password "robuste"
- impedire l'accesso esterno a questi servizi

 12

I servizi per la condivisione di file e directory in rete sono diffusissimi. **NetBios** e' il protocollo applicativo di Microsoft per gestire condivisioni in rete, altri meccanismi sono presenti in tutte le piattaforme (Unix/Linux, Apple, etc.). Una **NULL Session** e' il meccanismo usato su reti Windows per cercare informazioni sui computer della rete Microsoft (es. Quando siesplora la rete attraverso Windows Explorer). Esiste un account speciale SYSTEM usato per accedere a tali informazioni aprendo una NULL Session, una sessione di comunicazione speciale tra i computer che non richiede autenticazione.

I problemi sorgono da vulnerabilita' nell'implementazione dei meccanismi di condivisione o in coonfigurazione inappropriate. NetBios ha presentato molte di queste casistiche, cosi' come il meccanismo delle NULL Session. Cio' puo' provocare sia l'esposizione di file critici che la possibilita' di ottenere il controllo remoti di sistemi.

In molti casi poi, l'imprudente definizione di relazioni di fiducia ("trust relationships") ha effetti catastrofici. Relazioni di fiducia si instaurano quando si permette, esplicitamente e volontariamente, l'accesso e la condivisione di dati o servizi tra computer diversi. Tipicamente tra colleghi, dipartimenti, filiali, clienti, fornitori o partner aziendali.

Cio' puo' provocare:

-l'accesso da parte di utenti e organizzazioni non controllabili e la cui politica di sicurezza potrebbe imporre standard qualitativi inferiori rispetto quanto stabilito per l'organizzazione che mette a disposizione risorse;

-La condivisione di risorse tra computer aventi grado di rischio molto differente (es. Computer collocati all'interno di una rete protetta e non accedibile via Internet e computer accedibili via Internet o collocati sullo stesso ramo di rete di server che forniscono servizi Internet).

Internet Explorer, LAN Manager e Windows Scripting Host

Sistemi: Windows

- Internet Explorer, il browser Web di Microsoft, installato di default da Windoes presenta vulnerabilita' che possono essere sfruttate da un creatore di pagine web appositamente realizzate;

- LAN Manager e' un servizio di gestione di reti locali che viene installato di default in ambiente Windows anche quando il computer non necessita di tali funzionalita'.

- Windows Scripting Host (WSH) e' il meccanismo che permette l'esecuzione di script Visual Basic in maniera automatica. E' il meccanismo sfruttato spesso da virus e worm per la diffusione attraverso il mailer di posta Outlook o Outlook Express.

-**Contromisure minime**

- disabilitare le funzionalita' non indispensabili (es. LAN Mng., WSH);

- Installare patch e service pack

 13

Ancora, molti problemi di sicurezza derivano da funzionalita' avanzate abilitate nelle configurazioni standard di sistemi e applicazione di larghissima diffusione.

SSH, FTP, LPD e Sendmail

Sistemi: Unix/Linux

- SSH permette copia di file remota e consolle remota su un canale crittografato; FTP solo trasferimento di file;

- LPD e' il servizio piu' diffuso in sistemi UNIX/Linux per gestire la stampa. Sendmail invece e' il servizio piu' diffuso per la gestione dell'invio di posta elettronica.

- Tutti questi servizi, utilizzati in maniera amplissima, hanno presentato vulnerabilita' nelle diverse versioni che hanno fornito lo strumento per numerosissime intrusioni.

-**Contromisure minime**

- disabilitare le funzionalita' non indispensabili;

- Installare patch;

- Progettare l'architettura di rete e la dislocazione dei server in modo tale da confinare gli effetti di una intrusione.

 14

Molti problemi di sicurezza derivano da funzionalita' di larghissima diffusione le cui implementazioni contengono errori d progettazione (bug) tali da permere intrusioni.

Vulnerabilita' del tipo di quelle documentate per i servizi considerati in questa diapositiva sono presenti in TUTTI i servizi di rete e interattivi. Questi, ssh, ftp etc., vengono qui citati solo perche' sono tra i piu' diffusi, non perche' siano gli unici a presentare vulnerabilita'.

Buffer Overflow

Forse la vulnerabilita' piu' diffusa, piu' frequentemente usata. Spesso permette il controllo remoto come root/administrator.

Esempi di applicazioni che l'hanno evidenziata:

- AOL Instant Messenger
- Microsoft IIS
- IMAP
- Linuxconf
- Microsoft Windows 2000 Active X Control
- Microsoft NetMeeting e Outlook
- Microsoft SQL Server 2000 e Commerce Server 2000
- Internet Security Systems BlackICE e RealSecure

Etc….

 15

Buffer: un numero di blocchi di memoria contigui che mantengono piu' istanze di uno stesso tipo di dati.

Esempi di buffer: Buffer di Input/Output, locazioni di memoria che conservano i valori passati come Input (es. Da una form Web) o valori pronti per essere visualizzati o ritornati come risposte.

Buffer Overflow: I dati da memorizzare dei blocchi di memoria destinati al buffer superano la dimensione assegnata al buffer stesso. Normalmente, quando questo accade (ERRORE ESTREMAMENTE COMUNE), il programma o l'applicazione entra in uno stato inconsistente, il controllo passa quindi al sistema operativo che genera un errore e interrompe l'esecuzione.

 16

Il meccanismo di bufferizzazione (uso di buffer per gestire l'Input/Output) e' diffusissimo e utilizzato in una molteplicita' di casi da un sistema operativo.

Possiamo assumere, in prima approssimazione, che QUALUNQUE operazione di Input/Output effettuata da un sistema utilizzi tale meccanismo.

Esempio di buffer overflow

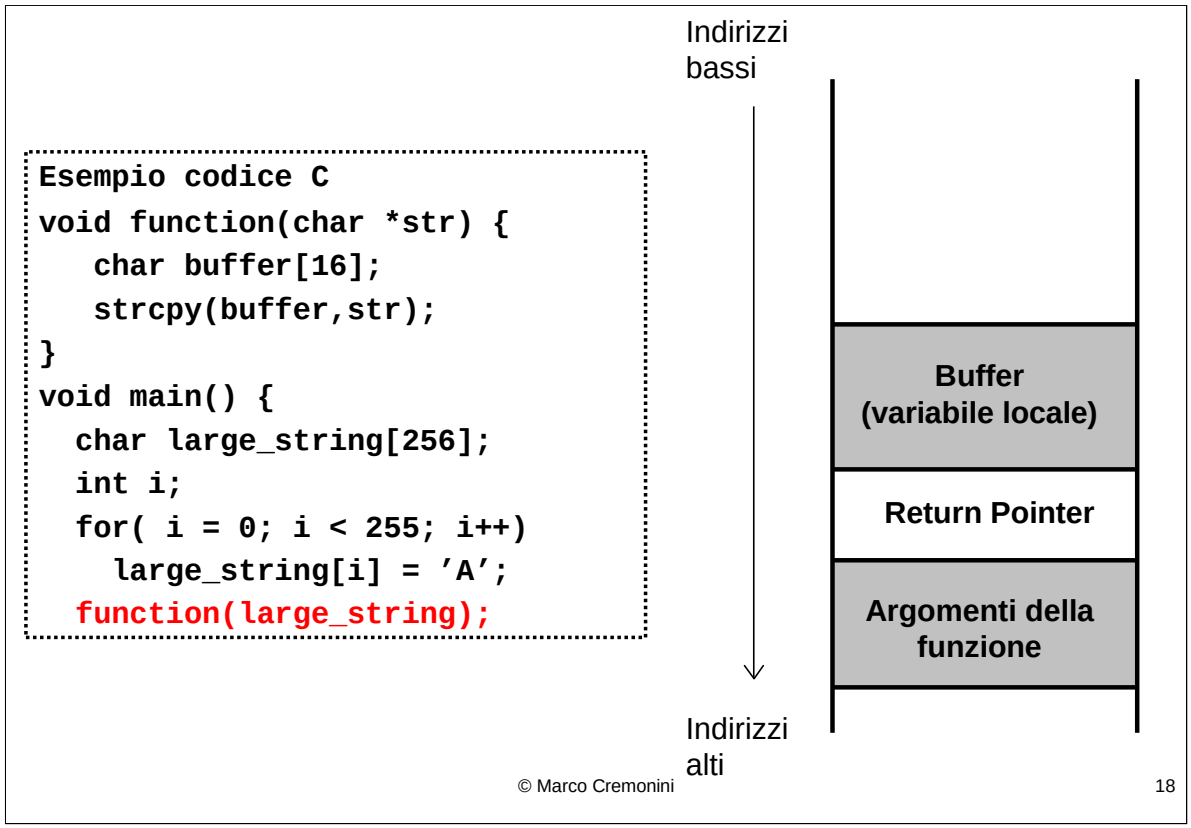
```
void function(char *str) {
   char buffer[16];
   strcpy(buffer,str);
}
void main() {
  char large_string[256];
  int i;
  for( i = 0; i < 255; i++)
    large_string[i] = 'A';
  function(large_string);
}
```

SEGMENTATION FAULT

SOLUZIONE: usare funzioni con controllo della dimensione dei dati,

es. **`strncpy()`** invece di **`srncpy()`**

17



Funzionamento normale dello stack.

1) Quando il programma chiama una funzione, gli argomenti della funzione vengono messi sullo stack;

2) Il puntatore (Return Pointer) va sullo stack;

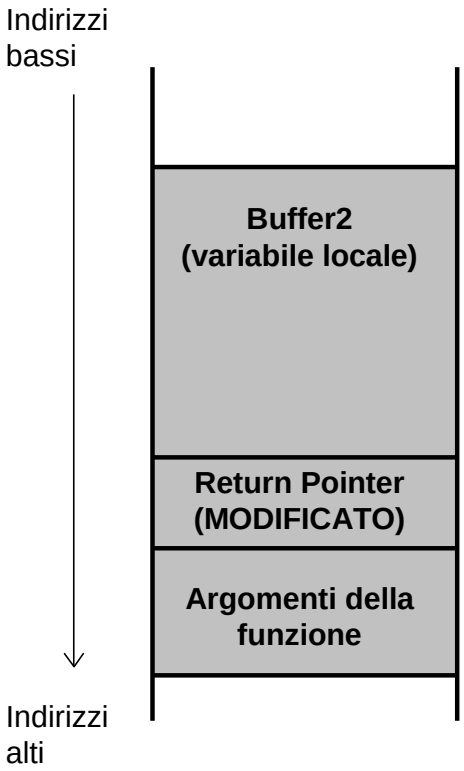
3) Le variabili locali alla funzione vanno, nell'ordine di definizione sullo stack.

Funzionamento LIFO (Last In First Out).

Il Return Pointer mantiene la locazione di memoria dell'istruzione chiamante nel flusso principale, prima che l'esecuzioni passi alla funzione.

SEGMENTATION FAULT: l'indirizzo di ritorno viene sovrascritto dagli "A", quando il processo esce dalla funzione e deve tornare al flusso principale, non trova un indirizzo corretto.

L'INDIRIZZO DI RITORNO E' STATO MODIFICATO

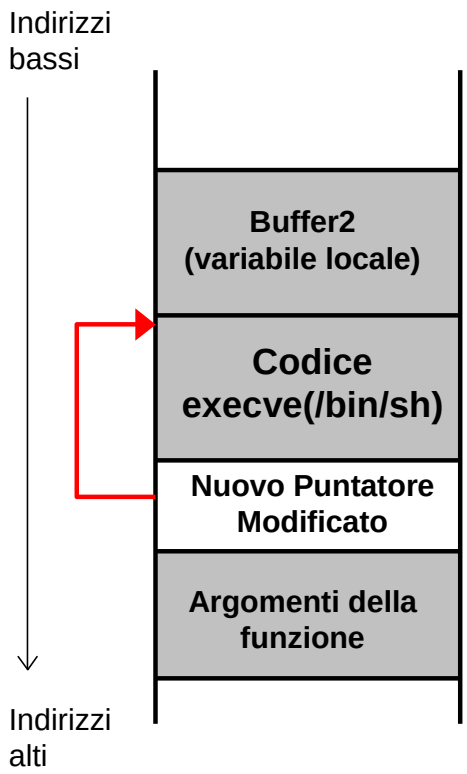


19

Stack in OVERFLOW.

PROBLEMA: Come si fa ad utilizzare in modo controllato l'overflow dello stack?

- ❑ Inserisco codice nei dati della variabile che va in overflow
- ❑ Faccio in modo che il nuovo valore del puntatore punti al codice stesso



 20

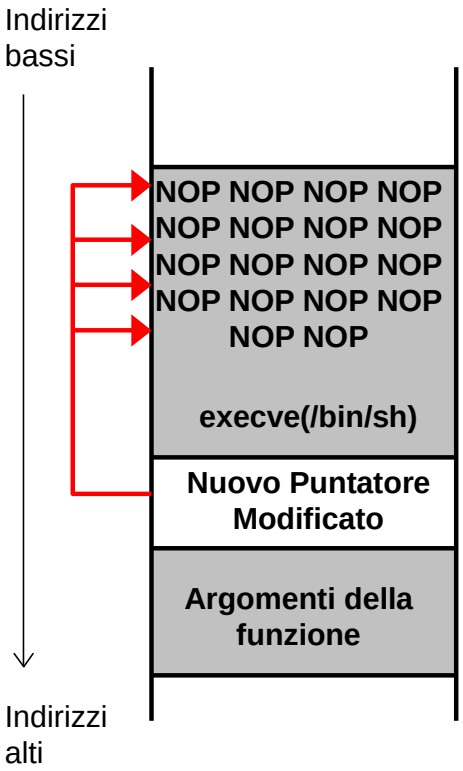
Non entriamo nei dettagli dei sistemi operativi.

Gli elementi da considerare per la discussione sono:

1. Una variabile (ad esempio associata all'input di un programma) riceve piu' dati della sua dimensione predefinita (overflow);
2. I dati in eccesso sovrascrivono altre locazioni di memoria non utilizzate nel sistema, al di fuori di quelle assegnate alla variabile;
3. All'interno dei dati di input sono inseriti comandi (es. Execve(/bin/sh/) che invoca una shell di sistema);
4. E' possibile (qui tralasciamo i dettagli) controllare la condizione di errore che l'overflow genera ed impedire che il sistema operativo assumendo il controllo interrompa l'esecuzione;
5. Al posto dell'interrozione dell'esecuzione, mediante un'opportuna configurazione dei dati di input, e' possibile indurre il sistema operativo (modifica del puntatore all'istruzione da eseguire) ad eseguire i comandi contenuti nell'input.

PROBLEMA: Come si fa a determinare ESATTAMENTE l'indirizzo di memoria allocato alla prima istruzione del codice?

- Estremamente difficile, dipende dall'implementazione dei singoli sistemi operativi
- Si rende piu' semplice inserendo molti istruzioni NOP (no operation), in modo da avere un insieme di locazioni di memoria da indirizzare con il puntatore.



Una delle difficalta' maggiori riguarda la determinazione della locazione di memoria contenente il comando passato nell'input e al quale il puntatore all'istruzione da eseguire deve riferirsi per evitare che il sistema operativo interrompa l'esecuzione.

Il tipico utilizzo di molte istruzioni NOP (istruzioni di No Operation, presenti in tutti i sistemi operativi che non hanno alcun effetto) permette a ci realizza un attacco di questo tipo di avere un certo margine di tolleranza nello stabilire la locazione del comando da far eseguire.

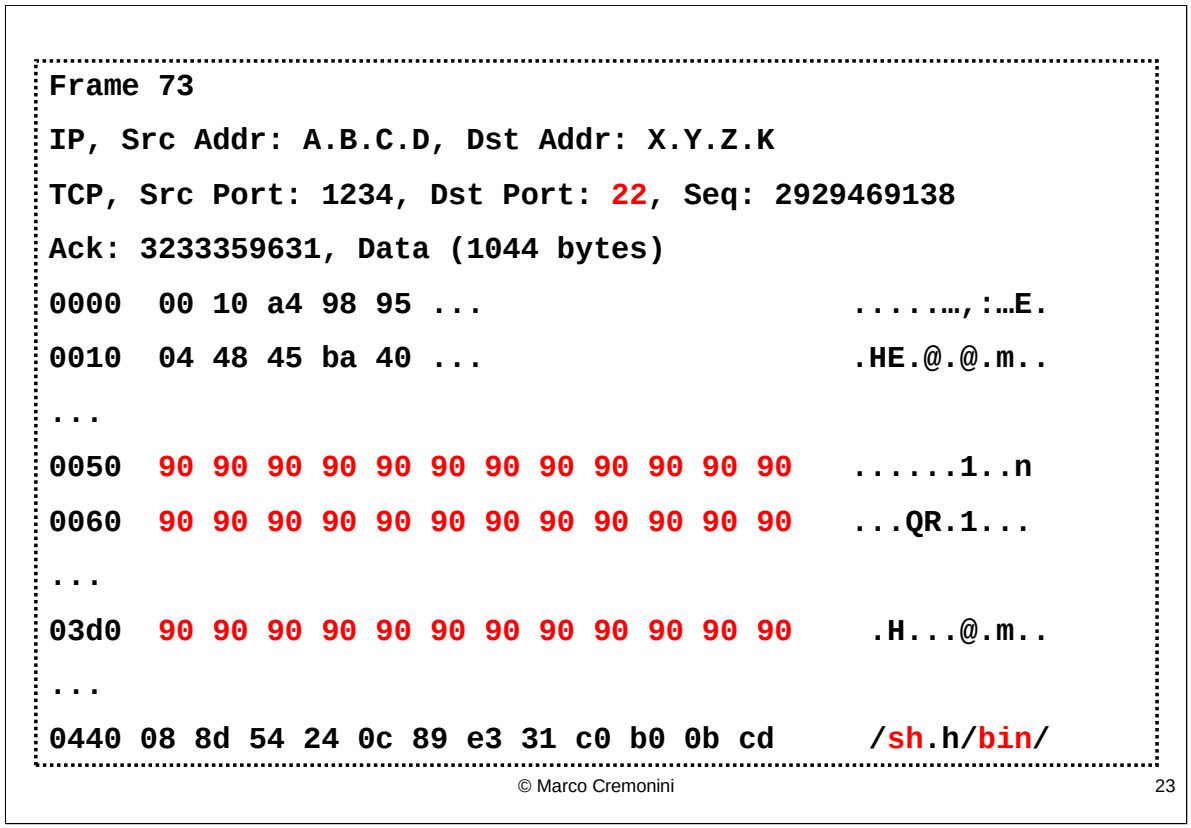
Il comando eseguito (es. /bin/sh) assumera' nella maggior parte dei casi i privilegi di root.

```
Linuxconf exploit 1999
...
char shell[]=
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90"
...
"…/bin/sh\x00";
...
printf("POST / HTTP / 1.0\r …);
<input di shell via POST http>
```

© Marco Cremonini 22

Esempio relativo a script che provoca buffer overflow sul servizio Linuxconf ed esegue /bin/sh per l'intrusore.

90 esadecimale (\x indica la notazione esadecimale) rappresenta il NOP.



Esempio di buffer overflow contro SSH (porta 22).

L'esempio mostra una traccia di pacchetto con visualizzazione esadecimale dei dati (payload).

(per motivi di spazio non sono stati riportati tutti i bytes nelle righe di dati)

NOTARE:

- la lunga sequenza di NOP (dal Byte 50 al Byte 984) (Hex 03d0 = DEC 976, ogni riga 8 bytes)
- l'invocazione del comando /bin/sh